APRENDE A PROGRAMAR

ANIMACIÓN

LA CASA ENCANTADA

ANAYA

PARA MSX Y SPECTRUM

Diseño: Cooper West
Edición: Nigel Cawthorne
Programas: Marcus Milton
David Rosam
Programas MSX Angel García García
Ilustraciones: Gerard Brown

Traducción: Angel García García

El traductor desea expresar su agradecimiento a
Gustavo Flores Maza por su colaboración en este libro

Queda prohibida la reproducción total o parcial de la presente obra bajo cualquiera de sus formas, gráfica o audiovisual, sin la autorización previa y escrita del editor, excepto citas en revistas, diarios o libros, siempre que se mencione la procedencia de las mismas.

Título original: COMPUTER ANIMATION
© 1985, Aladdin Books Ltd.
© 1986, de la edición española, E. G. Anaya.
Villafranca, 22. 28028 Madrid.
I.S.B.N.: 84-7525-361-X.
Depósito legal: M. 20984-1986
Impreso por: Edime, S. A. Calle D, esquina a F.
Polígono Industrial Arroyomolinos. Móstoles (Madrid)
Impreso en España - Printed in Spain.

APRENDE A PROGRAMAR

ANIMACIÓN

LA CASA ENCANTADA

Marcus Milton

ANAYA

Introducción

Los gráficos por ordenador encierran muchas más sorpresas de las que te puedes imaginar. En lugar de limitarte a jugar o ilustrar tus juegos puedes incluso escribir una historia, y construir tu propio programa lleno de color y gráficos tanto estáticos como animados. El programa se da para el sistema MSX y para el Spectrum y cuenta la historia de la Casa Encantada.

El programa ha sido dividido en bloques lógicos, y un texto paralelo explica cómo funciona cada sección. Escribir un programa en bloques lógicos no sólo lo hace más sencillo de entender tanto para ti como para otras personas; también reduce las posibilidades de error del mismo. Al finalizar determinadas secciones puedes probar lo que acabas de teclear; ten en cuenta que incluso el error más pequeño puede hacer que el programa no funcione. Si algo falla repasa los listados cuidadosamente. Asegúrate de que tu programa coincide con el del libro. Los siniestros acontecimientos que tienen lugar en la Casa Encantada emplean técnicas muy útiles de programación de gráficos. Por ejemplo, con un cuidadoso diseño de determinados caracteres definidos por el usuario en el Spectrum es fácil conseguir, mediante un programa en BASIC, un movimiento muy suave de las figuras.

Contenido

Gráficos MSX 8

Gráficos Spectrum 10

El escenario 12

PROGRAMA DE CONTROL E INICIALIZACION 13

MSX 14

SPECTRUM 16

DIA 19

MSX 20

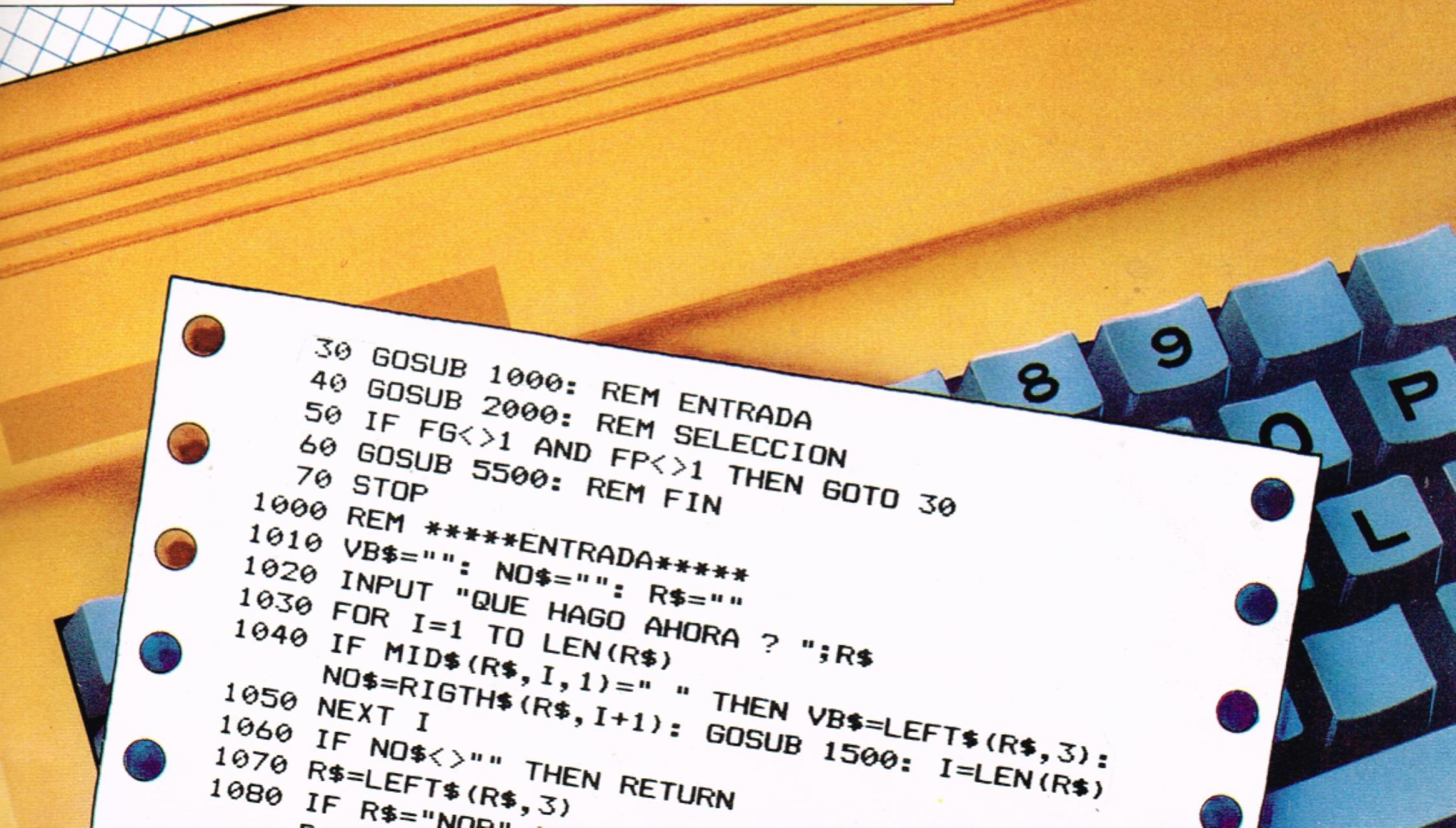
SPECTRUM 24

MEDIANOCHE 29

MSX 30

SPECTRUM 35

Listado de los programas 39



```
  30 GOSUB 1000: REM ENTRADA
  40 GOSUB 2000: REM SELECCION
  50 IF FG<>1 AND FP<>1 THEN GOTO 30
  60 GOSUB 5500: REM FIN
  70 STOP
1000 REM *****ENTRADA*****
1010 VB$="": NO$="": R$=""
1020 INPUT "QUE HAGO AHORA ? ";R$
1030 FOR I=1 TO LEN(R$)
1040 IF MID$(R$,I,1)=" " THEN VB$=LEFT$(R$,3):
     NO$=RIGTH$(R$,I+1): GOSUB 1500: I=LEN(R$)
1050 NEXT I
1060 IF NO$<>"" THEN RETURN
1070 R$=LEFT$(R$,3)
1080 IF R$="NOR"
```

Gráficos MSX

EL BASIC del MSX permite animar figuras de una manera muy simple mediante el uso de **sprites.** Estos son caracteres cuya forma podemos definir y mover por la pantalla. Podemos definir cuatro tipos de **sprites:** 8×8, 8×8 aumentado, 16×16 y 16×16 aumentado, según el número de pixels (puntos de la pantalla) y el tamaño de éstos.

Veamos sobre un ejemplo cómo definir un sprite y cómo moverlo por la pantalla.

```
10 SCREEN 2,2
20 FOR I=1 TO 32:READ N:A$=A$+CHR$(N):NEXT
30 SPRITE$(1)=A$
40 FOR J=240 TO 1 STEP -1:PUT SPRITE 1,(J,1):
   FOR I=1 TO 50:NEXT:NEXT
50 GOTO 40
60 DATA 0,0,0,112,120,60,30,127,255,255,127,1,0,0,0,0,
   0,0,0,4,14,30,62,254,255,255,254,254,240,120,60,28
```

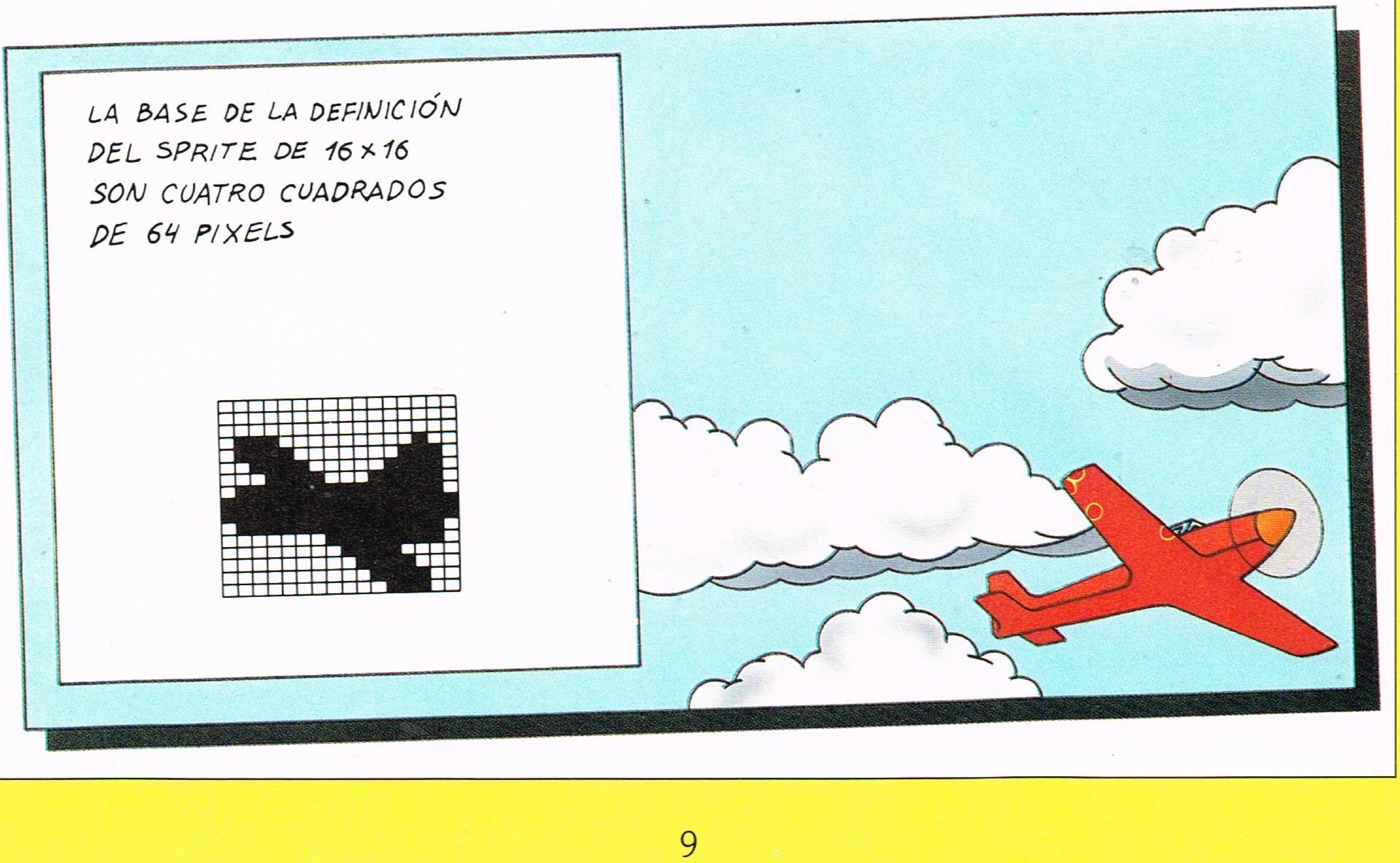
La línea 10 selecciona el modo de pantalla 2 —modo gráfico de 192×256 pixels— y los sprites de tipo 16×16. En las líneas 20 y 30 definimos el sprite leyendo los 32 datos necesarios (pues para el sprite de 16×16 necesitamos 32 bytes) y asignando una cadena de caracteres a la variable **SPRITE** $. El argumento de esta variable (1 en este caso) selecciona el plano de definición de ese sprite. Para entenderlo imaginemos la pantalla del monitor o televisor compuesta por una serie de capas (PLANOS). Las imágenes en los planos más cercanos al observador —planos inferiores— parecen pasar por delante de las de los que se encuentran en planos posteriores. La línea 40 se encarga de mover el sprite por la pantalla; para ello indicaremos, en este orden, de qué sprite se trata, sus coordenadas y el color en que queremos que se imprima.

Después de los planos de sprites se encuentra el “plano multicolor”. En él es donde aparecen las imágenes creadas por manejo de textos y los comandos gráficos usuales **PRINT**, **DRAW**, **LINE**... No obstante podemos modificar las imágenes que aparecen en este plano para crear figuras más sofisticadas como los árboles que veremos en la Casa Encantada.

Para ello hay que acceder directamente a la memoria de vídeo del ordenador. El proceso es relativamente complejo y aquí sólo pretendemos dar una idea general del mismo. Toda imagen en la pantalla viene definida por las posiciones de memoria de la **VRAM** (memoria de vídeo). En el modo 2 (gráfico) ésta se divide en cinco bloques: **tabla de patrones,** que nos da la definición de cada carácter en la pantalla de forma análoga a la de los sprites; **tabla de nombres,** que nos indica para cada posición de pantalla dónde se encuentra en la tabla de patrones la definición de dicha posición; **tabla de colores,** que nos da el color de fondo y de primer término de cada byte —8 pixels horizontales— de la tabla de patrones; **tabla de patrones de sprites,** sobre la que podemos actuar directamente para definir los sprites y, por fin, la **tabla de atributos de sprites,** que nos da la posición, color y número de patrón de cada sprite.

Veamos el proceso que hay que seguir para crear una imagen en el plano de fondo. En primer lugar definiremos en la tabla de patrones la figura que pretendemos obtener de una forma análoga a la de definición de sprites. A continuación indicaremos en la tabla de nombres para las posiciones de carácter que pretendemos que tengan esa forma que el patrón se encuentra a partir de un lugar determinado de la tabla de patrones. Por último, si queremos, podemos modificar el color de cada carácter actuando sobre la tabla de colores.

Aunque el proceso pueda parecer bastante complejo, si lo manejamos con cuidado podemos llegar a obtener resultados sorprendentes. No obstante conviene acudir a un libro especializado para profundizar debidamente en el tema.



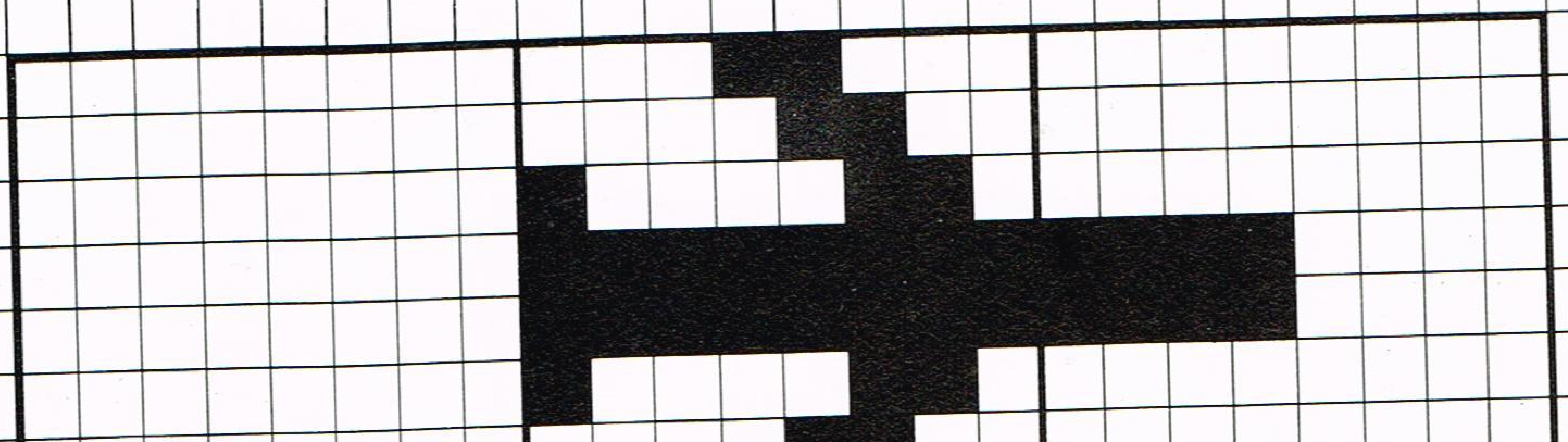
Gráficos Spectrum

La forma usual de animar un cáracter desde el BASIC en el Spectrum es imprimir la figura, borrarla y volverla a imprimir en una posición contigua. Generalmente el efecto de movimiento no es muy bueno, pues la figura parece moverse a saltos. Y lo que es peor, las figuras no pueden moverse uno o dos pixels cada vez (deben localizarse en filas y columnas). No obstante, si imprimimos tres o cuatro formas apropiadas en el espacio de dos caracteres podemos crear la ilusión de que la figura se está moviendo sólo unos pocos pixels cada vez. Podemos conseguir un movimiento suave, libre de saltos, repitiendo este proceso sobre posiciones sucesivas de la pantalla.

Prueba este programa.

```
 10 FOR A=65368  TO 65439 :READ D: POKE A,D: NEXT A
 20 DATA 24,12,134,255,255,134,12,24: REM (A)
 30 DATA 0,0,0,240,240,0,0,0: REM (B)
 40 DATA 6,3,33,63,63,33,3,6: REM (C)
 50 DATA 0,0,128,252,252,128,0,0:REM (D)
 60 DATA 1,0,8,15,15,8,0,1: REM (E)
 70 DATA 128,192,96,255,255,96,192,128:REM (F)
 80 DATA 0,0,2,3,3,2,0,0: REM (G)
 90 DATA 96,48,24,255,255,24,48,96: REM (H)
100 DATA 0,0,0,192,192,0,0,0:REM (I)
```

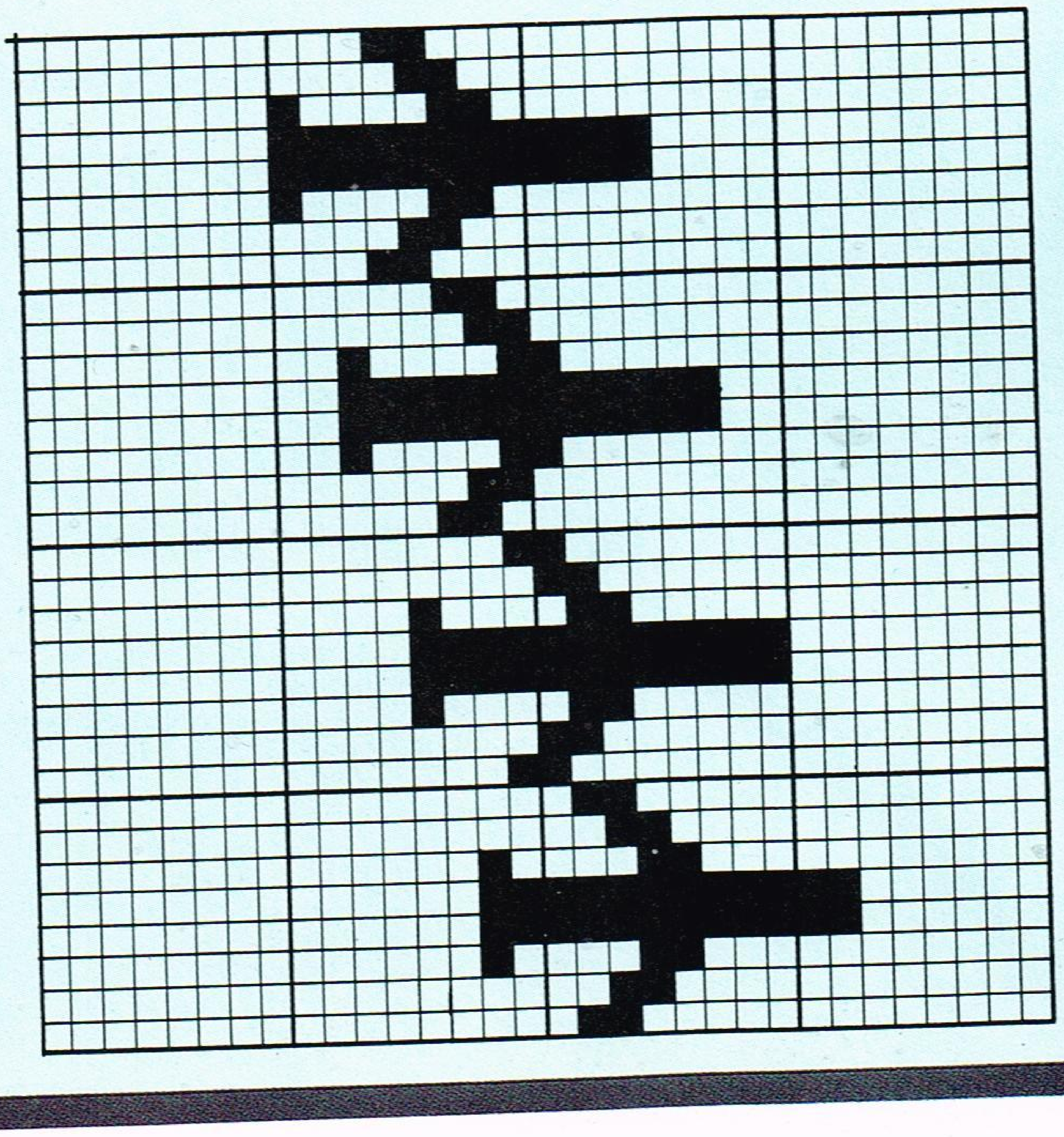
UNA VEZ QUE HAYAS INTRODUCIDO Y EJECUTADO EL PROGRAMA, TECLEA : **PRINT AT 2,0;" ↦— "** PULSA ENTER , EL AVIÓN APARECERÁ EN LA PARTE SUPERIOR IZQUIERDA DE LA PANTALLA. RECUERDA QUE LOS SÍMBOLOS ↦ Y — SE CONSIGUEN PONIENDO EL ORDENADOR EN MODO GRÁFICO Y PULSANDO LAS TECLAS **A** Y **B** RESPECTIVAMENTE



```
110 FOR P=0 TO 28
120 PRINT AT 2,P;" AB": PAUSE 3
130 PRINT AT 2,P;" CD": PAUSE 3
140 PRINT AT 2,P;" EF": PAUSE 3
150 PRINT AT 2,P;" GHI": PAUSE 3
160 NEXT P
```

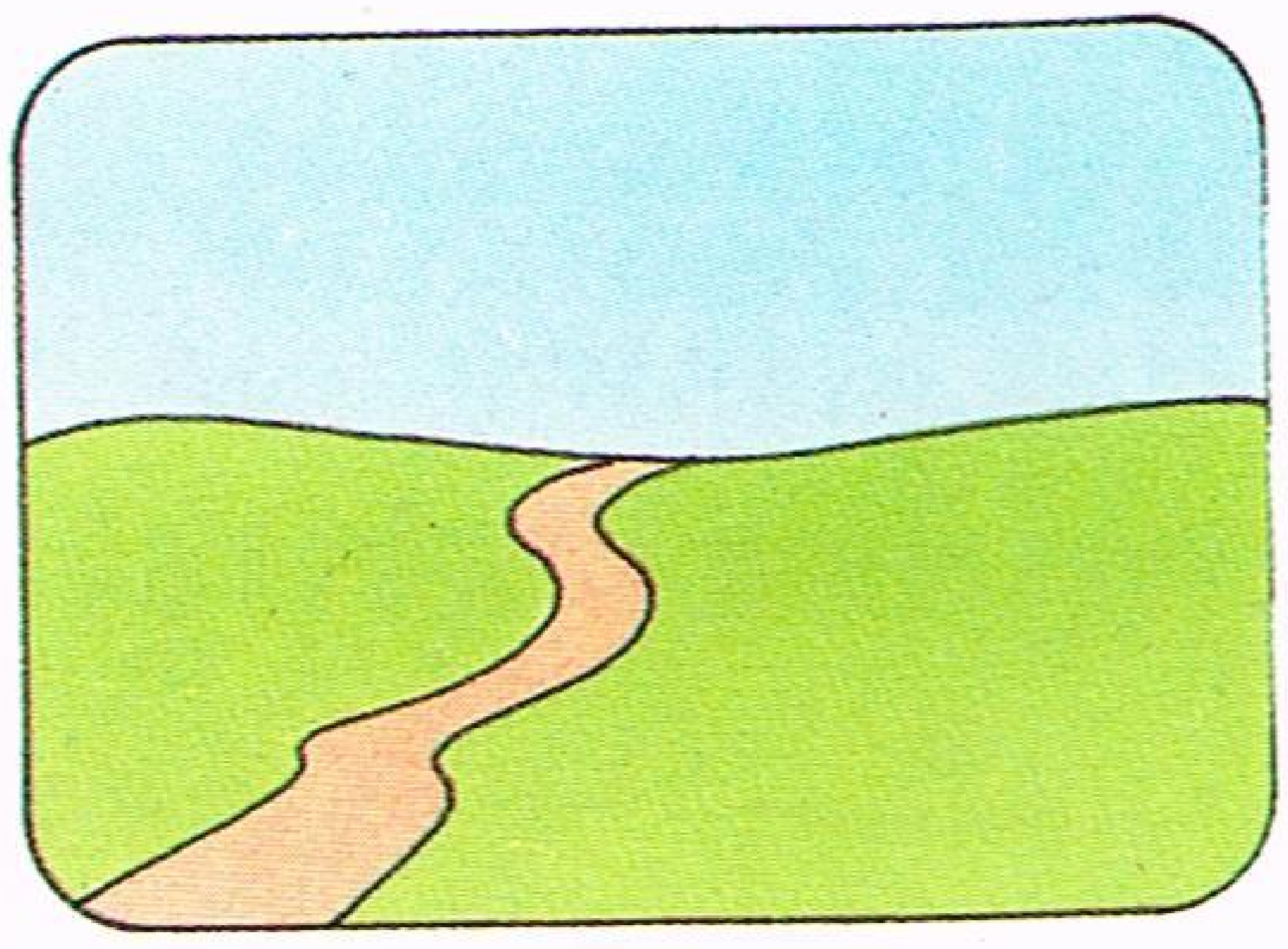
Las líneas 110 a 160 mueven el avión suavemente por la pantalla. El secreto es que en realidad hay cuatro gráficos que parecen iguales, excepto porque están situados en posiciones diferentes. Cada gráfico se construye con dos o tres caracteres definidos por el usuario (líneas 120 a 150). La figura que se imprime en la línea 130 está localizada dos pixels a la derecha de la figura impresa en la línea 120; la figura de la línea 140 está situada dos pixels a la derecha de la de la línea 130, y así sucesivamente. El lazo entre las líneas 110 y 160 mueve el avión por la pantalla. El conjunto de caracteres definidos por el usuario que se imprimen en las líneas 120 a 150 es como una minisecuencia animada (al imprimir rápidamente la secuencia de gráficos de avión conseguiremos que éste parezca moverse incluso dentro de un sólo carácter). El lazo se limita a presentar en pantalla esta minisecuencia animada en cada carácter de la línea. El carácter que imprimimos justo antes de cada grupo de caracteres borra la última parte de la cola del avión, que de no ser así quedaría atrás según se mueve éste por la pantalla.

EL MOVIMIENTO DEL AVIÓN SE CONSIGUE IMPRIMIENDO LA SERIE DE GRÁFICOS DE LA DERECHA. TODAS LAS FIGURAS SON IGUALES PERO COLOCADAS EN UNA POSICIÓN DIFERENTE

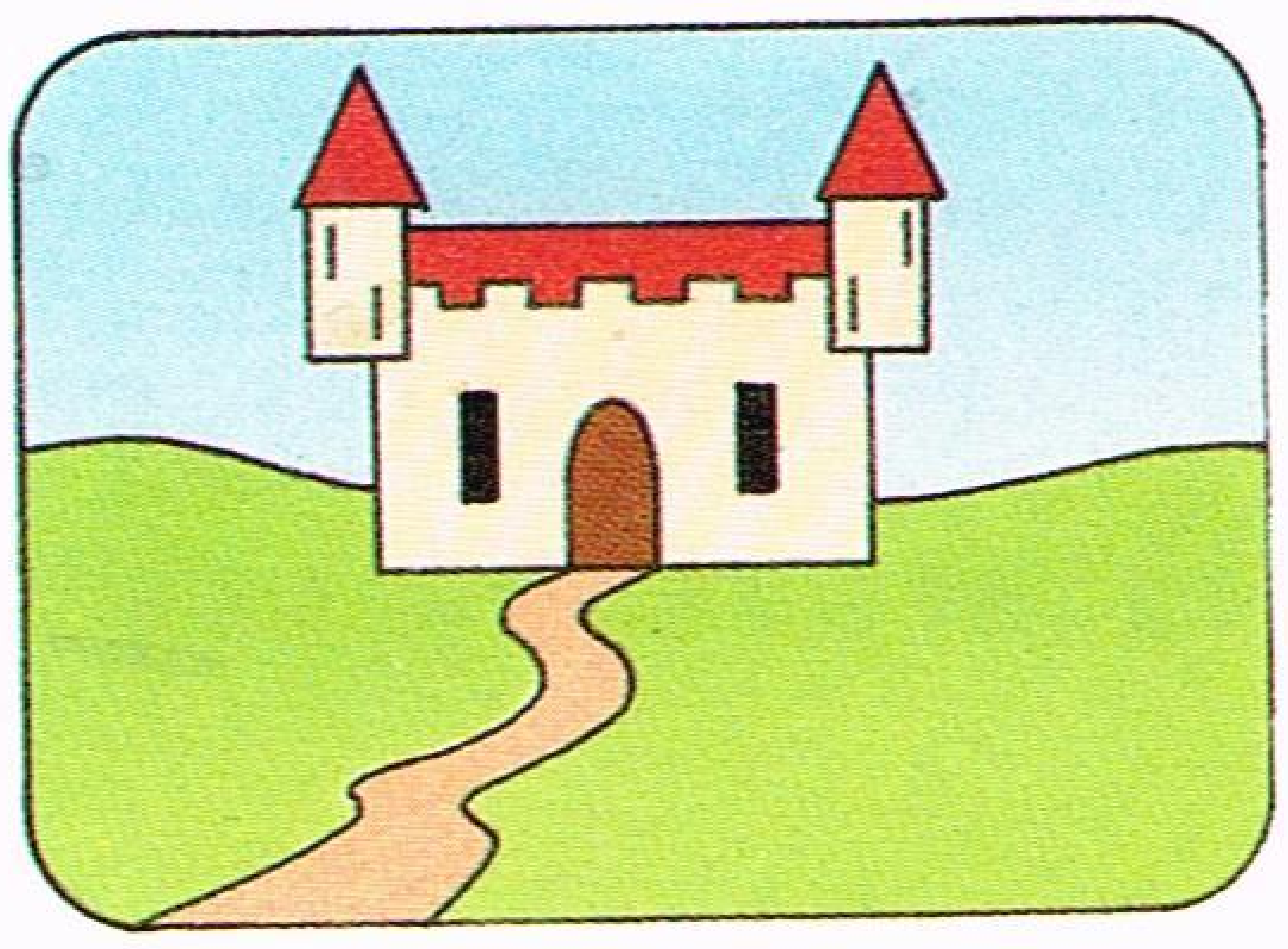


El escenario

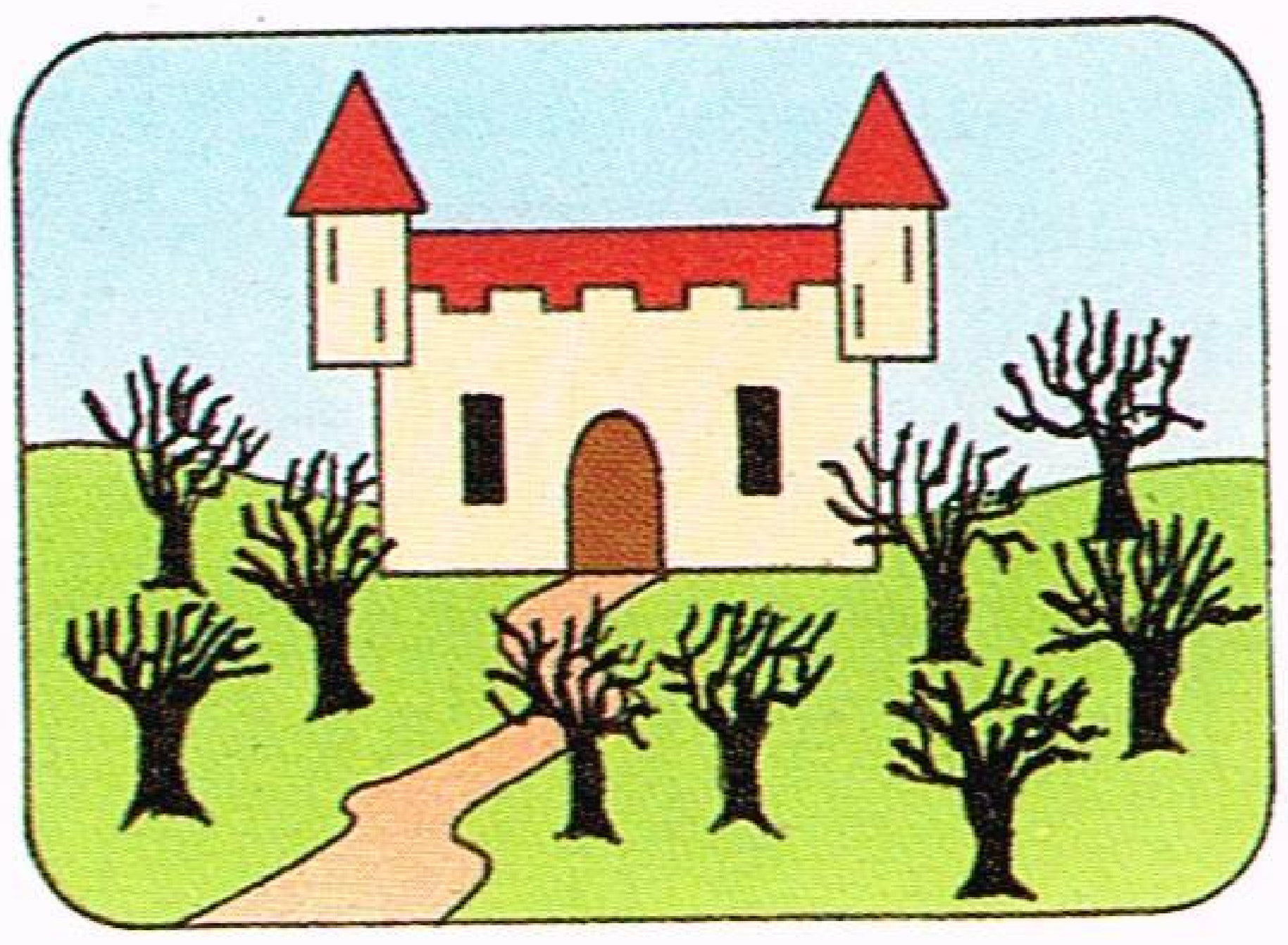
Antes de comenzar con el programa tenemos que elaborar el argumento o guión de nuestra historia. El programa da vida a una casa encantada y comienza por dibujar la imagen de una casa antigua rodeada de un bosque fantasmagórico. Cae la noche... un rayo atronador despierta a los murciélagos del campanario y una vieja bruja sobrevuela la casa en su escoba. Mientras vuela en la noche caen más rayos y relámpagos, que alcanzan la casa y provocan el fuego en una de las torres. Las llamas se extienden y la casa entera se derrumba. Como dramática conclusión de la historia una tétrica calavera se alza sobre sus ruinas. La secuencia dura unos minutos. Conviene dibujarse un guión como el de abajo en un papel, de forma que puedas dividir el argumento en escenas y sucesos separados.



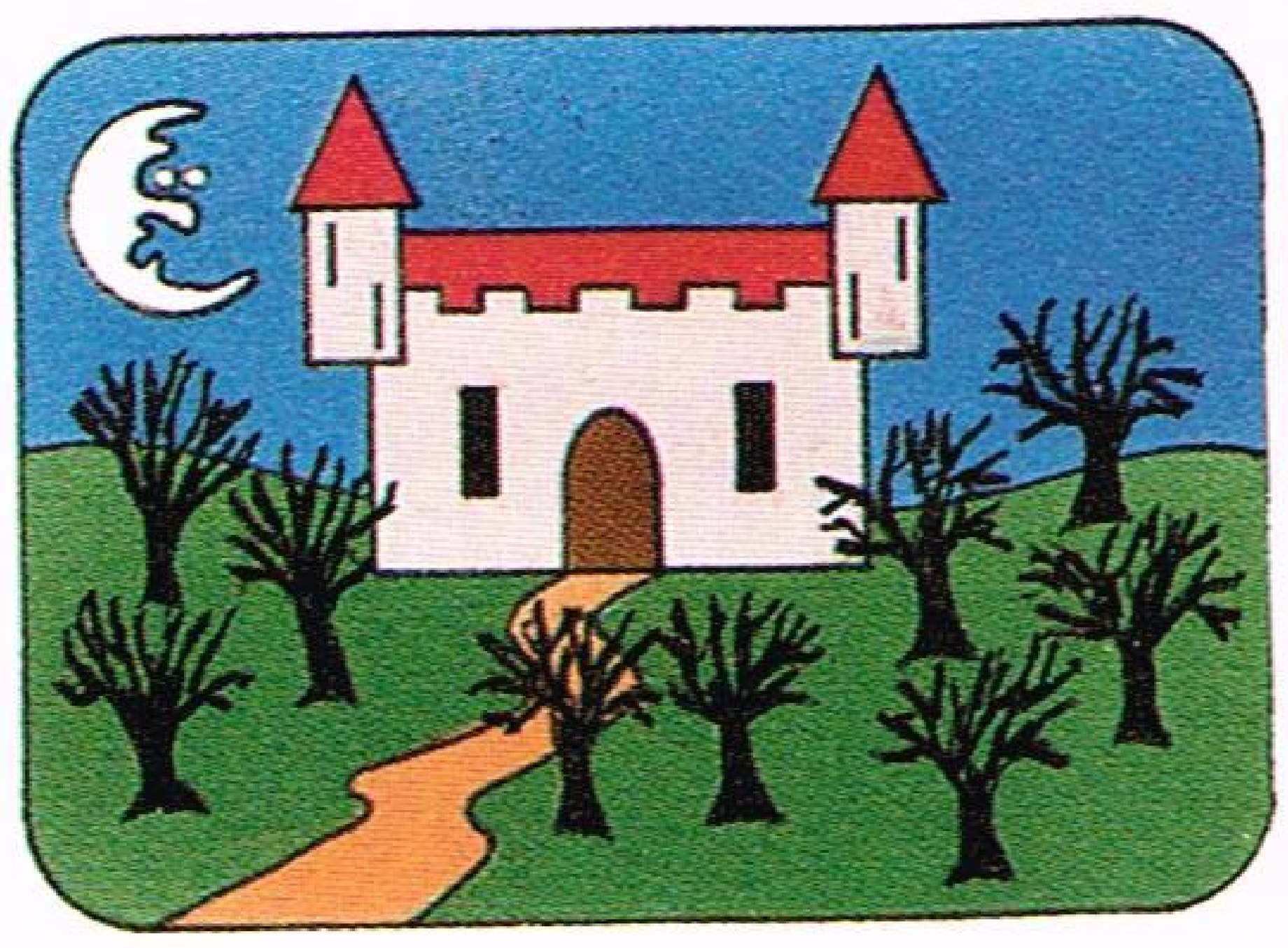
FONDO



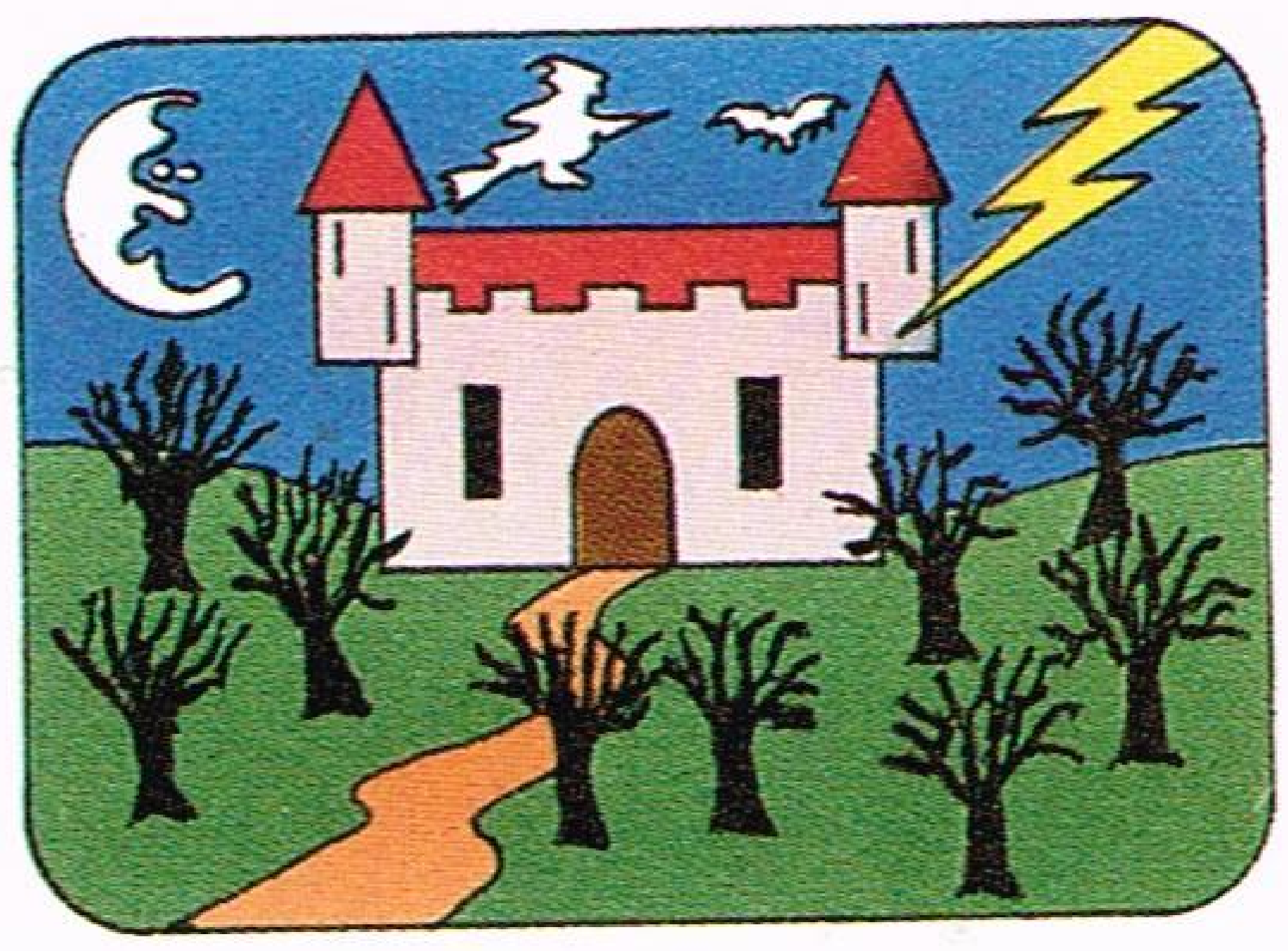
CASA ENCANTADA



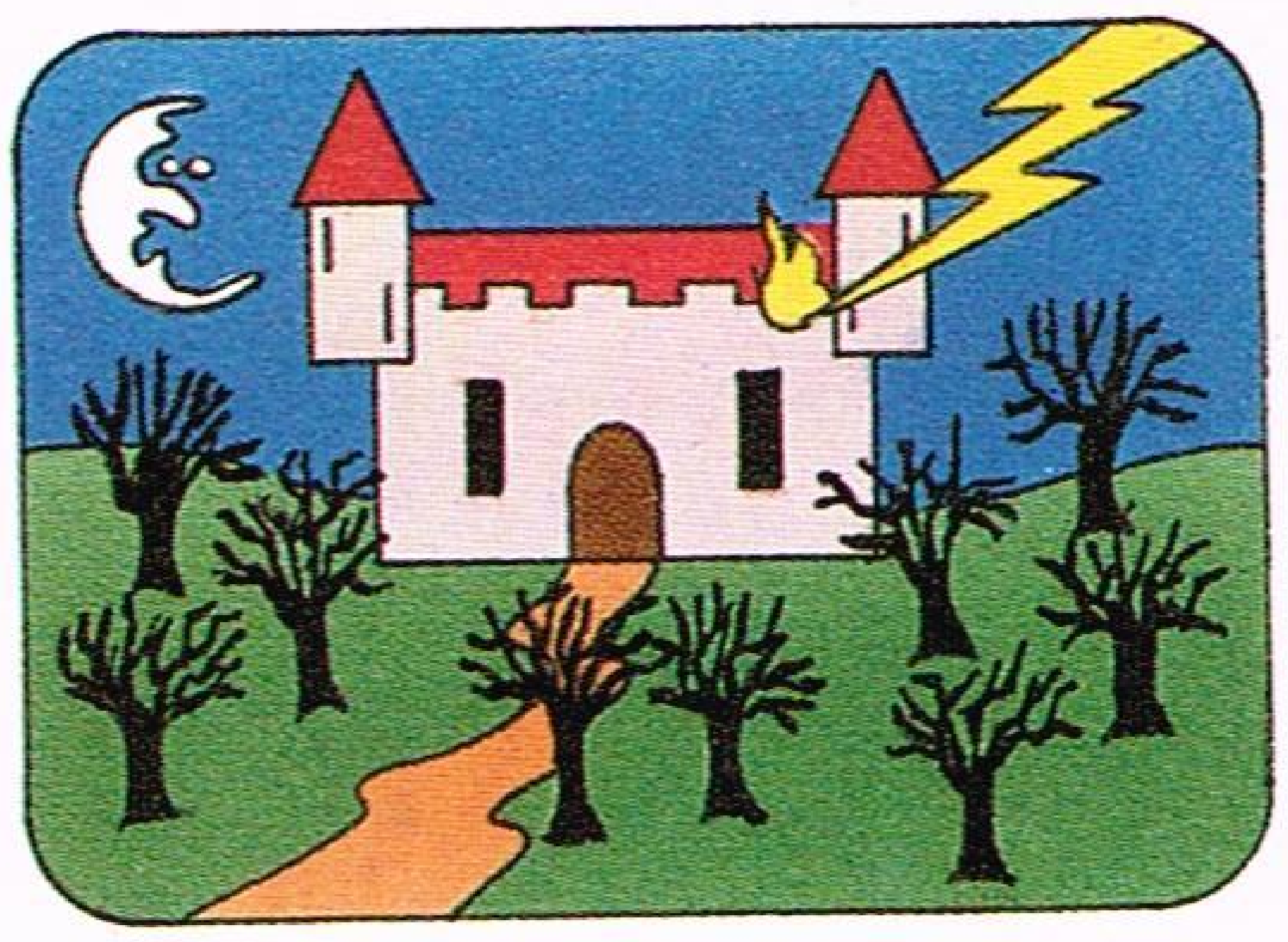
BOSQUE ENCANTADO



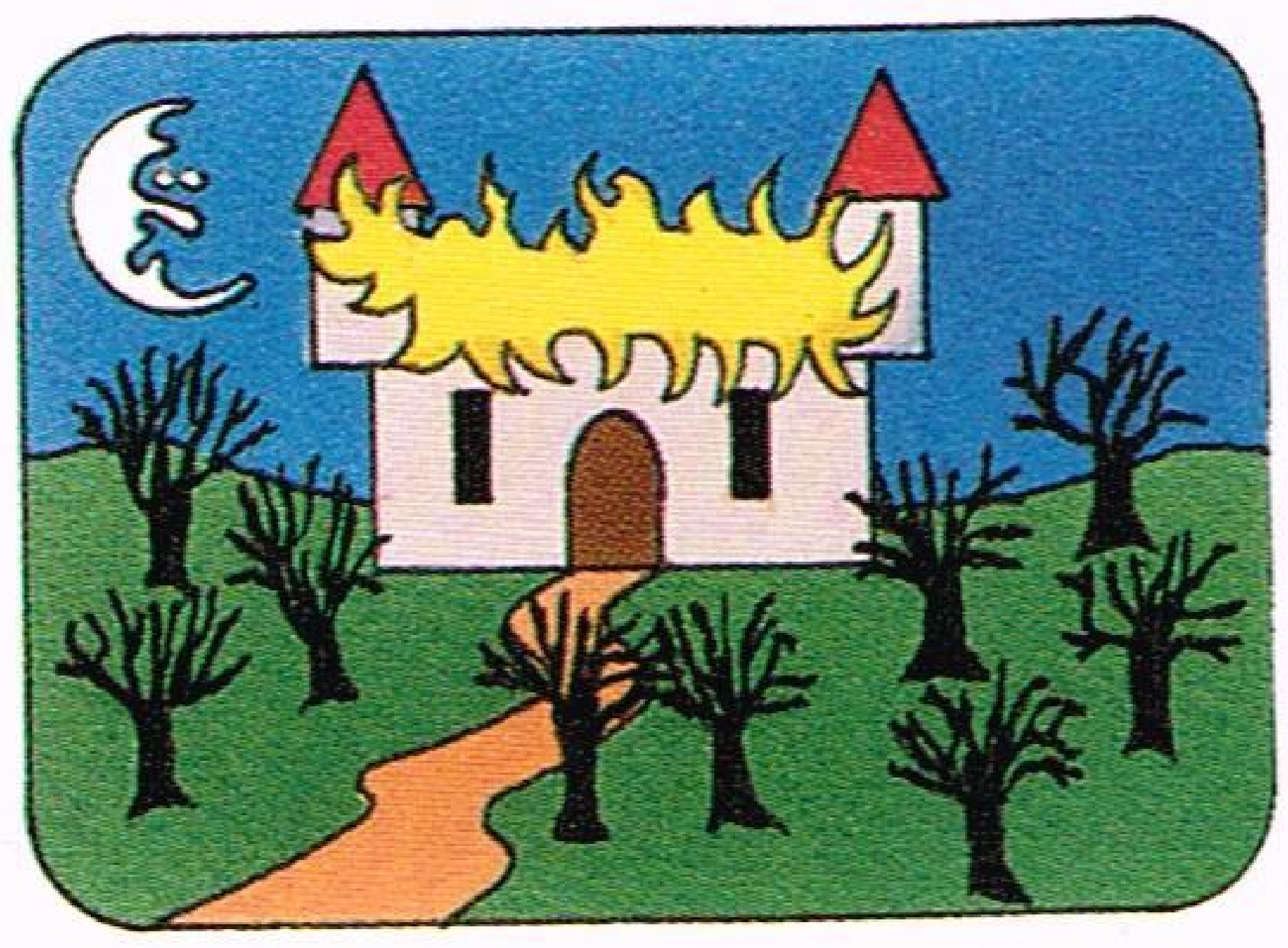
CAE LA NOCHE



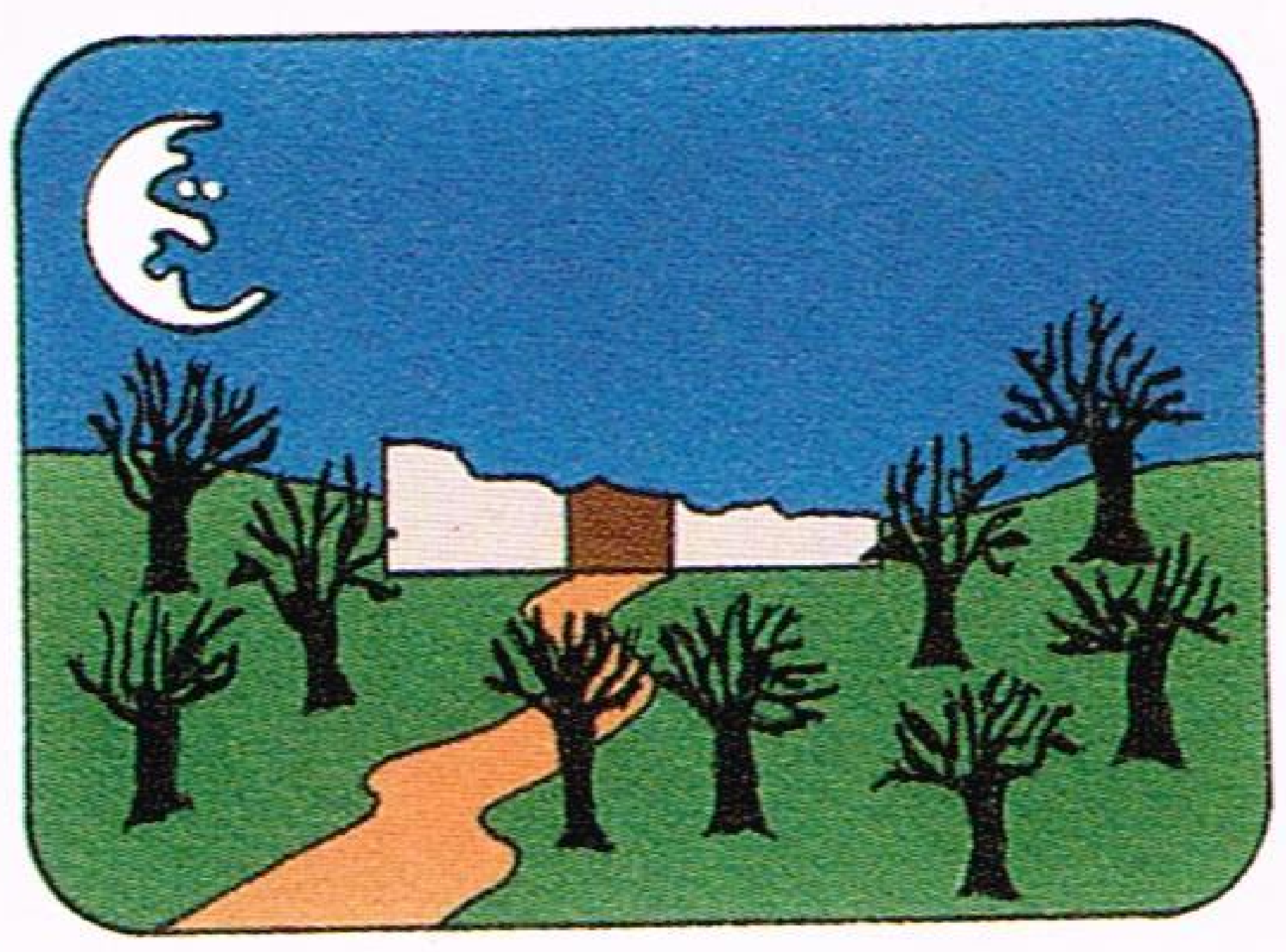
MURCIÉLAGO Y BRUJA



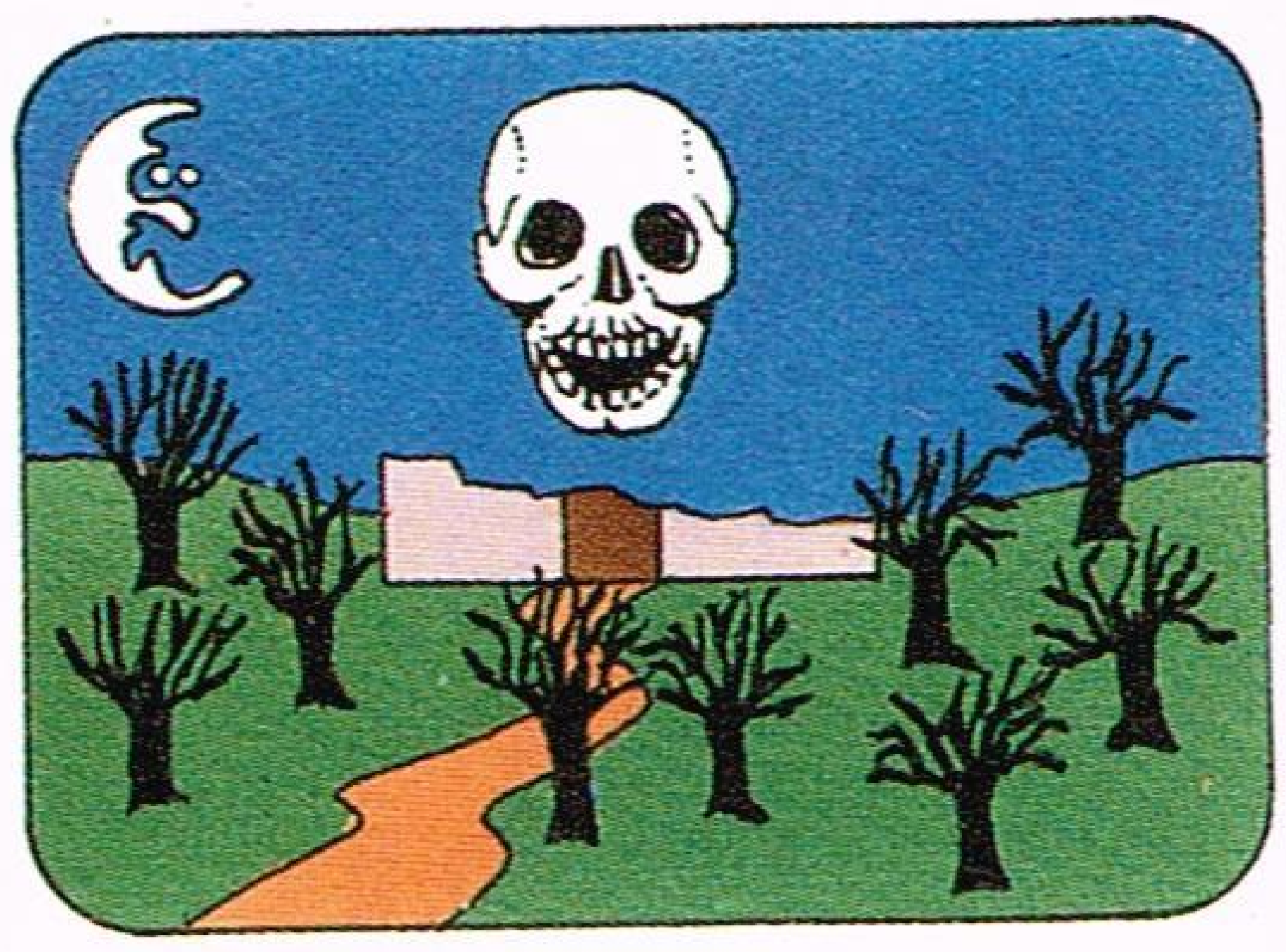
EL RAYO ALCANZA LA CASA



LA CASA ARDIENDO



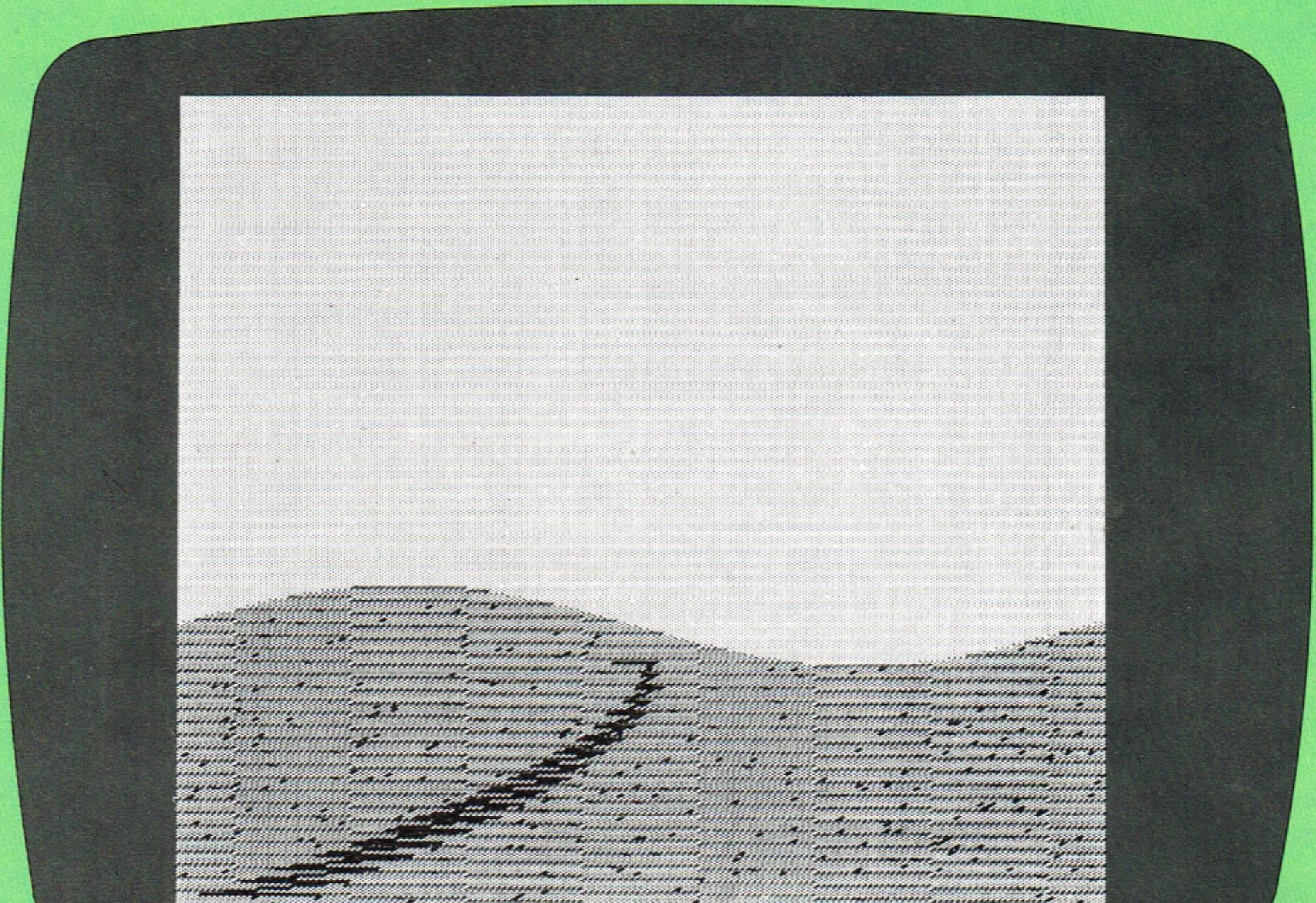
LA CASA SE DERRUMBA



APARECE LA CALAVERA

EL PROGRAMA DE CONTROL Y LA INICIALIZACION

Una vez que hayas planeado la historia puedes escribir el programa de control. Este te permite estructurar tu programa, dividiendo cada sección en una serie de bloques más pequeños y manejables. Esto lo conseguirás mediante el empleo de subrutinas. La primera sección te muestra cómo escribir el programa de control y preparar el ordenador para el comienzo de la animación.



La estructura del programa de control se ajusta al argumento. La mayoría de las subrutinas se utilizan una sola vez, como las empleadas para dibujar el castillo y los árboles. Las únicas subrutinas que se repiten son las llamadas en las líneas 100 y 110. Con ellas conseguimos mover el murciélago y la bruja por la pantalla.

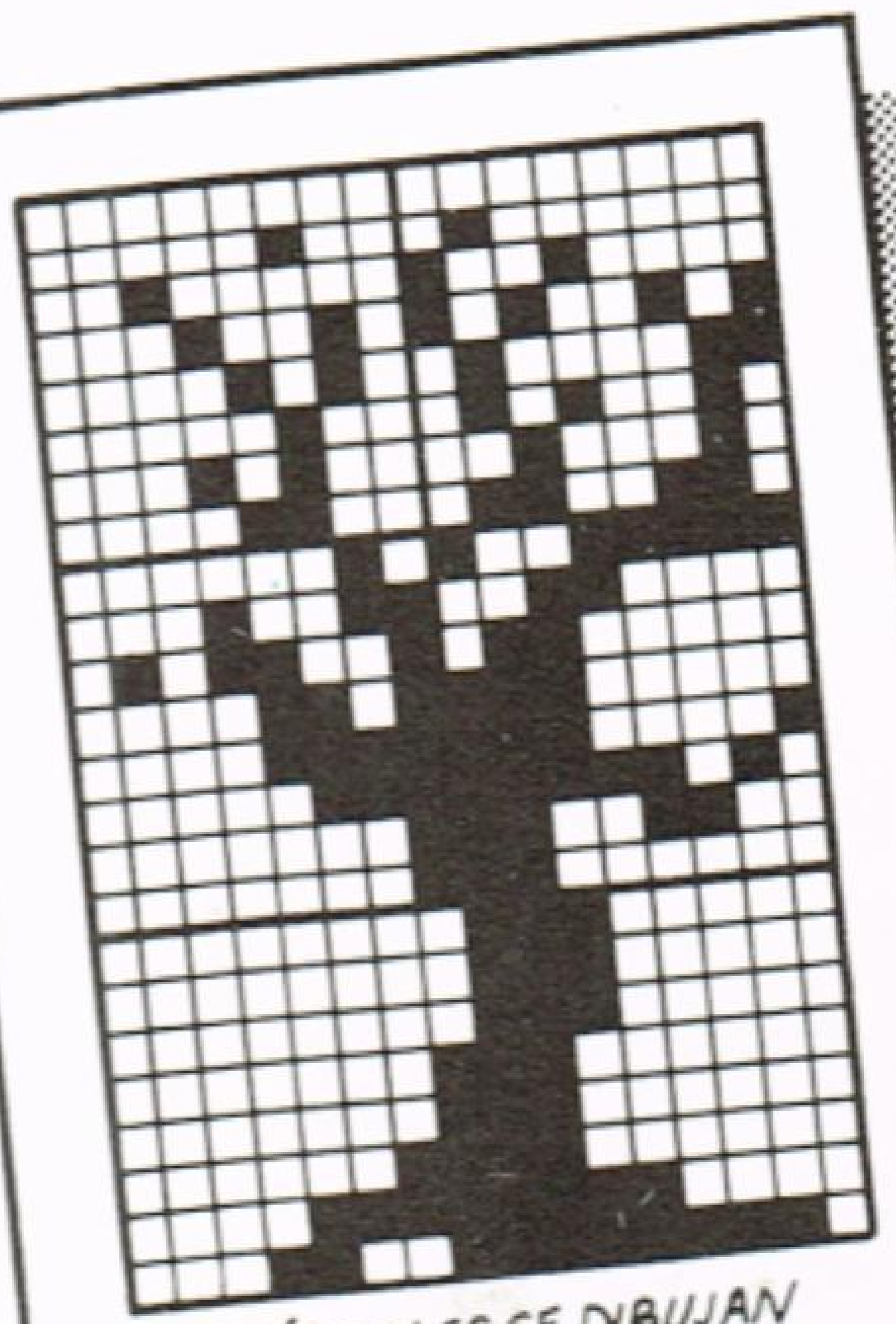
```
   1  REM Casa Encantada
   5  CLEAR 200,61482!
  10  SCREEN 2,2:COLOR 1,3,1:CLS
  20  GOSUB 6500:REM Inicializar
  30  GOSUB 2000:REM Fondo
  40  GOSUB 2500:REM Castillo
  50  GOSUB 4500:REM Arboles
  60  GOSUB 5000:REM Oscuridad
  70  N=4:GOSUB 1500:REM Rayo
  80  M=1300
  90  FOR XB=0 TO 254 STEP 2
 100  GOSUB 500:REM Murcielago
 110  GOSUB 1000:REM Bruja
 120  IF XB=254 THEN PUT SPRITE 3,(0,209)
 130  NEXT XB
 140  N=4:GOSUB 1500:REM Rayo
 150  GOSUB 5500:REM Fuego
 160  GOSUB 6000:REM Calavera
 170  END
```

El programa de control se limita a llamar consecutivamente a una serie de subrutinas para conseguir el dibujo en la pantalla. En la línea 70, antes de llamar a la subrutina **Rayo** establecemos el valor de **N** que será empleado por ésta para determinar el número de relámpagos. Asimismo la línea 80 actúa sobre la variable **M** necesaria en la subrutina **Murciélago.** Es conveniente trabajar de esta forma con las subrutinas, pues así se hacen mucho más flexibles.

```
6500 REM Inicializar
6510 FOR I=61483! TO 61508!
6520 READ N$:POKE I,VAL("&H"+N$):NEXT
6530 DEFUSR=61483!
6540 FOR I=BASE(12) TO BASE(12)+15:READ N:
     VPOKE (I+4096),N:NEXT
6550 FOR I=BASE(12) TO BASE(12)+15:READ N:
     VPOKE (I+4352),N:NEXT
6560 FOR I=BASE(12) TO BASE(12)+15:READ N:
     VPOKE (I+4608),N:NEXT
6570 FOR I=BASE (14)+32 TO BASE(14)+191:READ N:
     VPOKE I,N:NEXT
6580 RETURN
```

```
7000 DATA DB,99,21,00,20,0E,99,ED,69,ED,61,21,00,00,3E,
     14,D3,98,23,7C,FE,18,C2,39,F0,C9:REM Cod. maquina
7010 DATA 16,129,73,40,16,80,48,9,128,32,74,134,164,68,
     204,30,5,38,178,27,31,15,3,3,48,96,192,192,194,244
     ,152,128,1,1,1,3,3,7,31,51,192,192,192,128,128,128
     ,224,252:REM Arbol
7020 DATA 0,0,0,16,56,125,255,63,15,3,2,0,0,0,0,0,0,0,0
     ,8,28,190,255,252,240,192,64,0,0,0,0,0:REM Murc.1
7030 DATA 0,0,0,0,0,13,31,63,63,63,26,24,8,0,0,0,0,0,0,
     0,0,176,248,252,252,252,88,24,16,0,0,0:REM Murc.2
7040 DATA 7,1,0,0,0,1,3,159,207,255,195,129,1,0,0,0,128
     ,192,96,96,224,192,192,224,240,255,192,128,128,192
     ,0,0:REM Bruja
7050 DATA 3,14,28,56,122,120,254,255,255,248,248,96,192
     ,55,31,3,0,0,0,0,128,0,0,128,128,0,0,4,12,152,248,
     240:REM Luna
7060 DATA 0,32,32,34,50,51,51,127,125,236,228,100,53,31
     ,7,7,0,0,17,17,49,115,227,167,167,191,186,164,200,
     208,224,224:REM Llama
7070 DATA 521,535,540,547,550,562,565,570,580,592,600,
     610,637,640,657,661,667,686,691,696,700:
     REM Posicion arboles
7080 DATA 177,48,169,64,149,84,91,84,71,64,63,48:
     REM Posicion llamas
```

La subrutina **Inicializar** comienza en la línea 6500. Las líneas 6510 y 6520 se encargan de leer los datos necesarios para configurar un programa en código máquina que emplearemos para cambiar instantáneamente los colores de la pantalla. La línea 6530 fija la dirección a donde queremos llamar al comando **USR**. Las líneas 6540 a 6560 establecen la definición del árbol accediendo a la tabla de patrones y actuando sobre la misma. La línea 6570 modifica las posiciones correspondientes a la tabla de patrones de sprites generando así los correspondientes al murciélago (dos), la bruja, la luna y la llama.



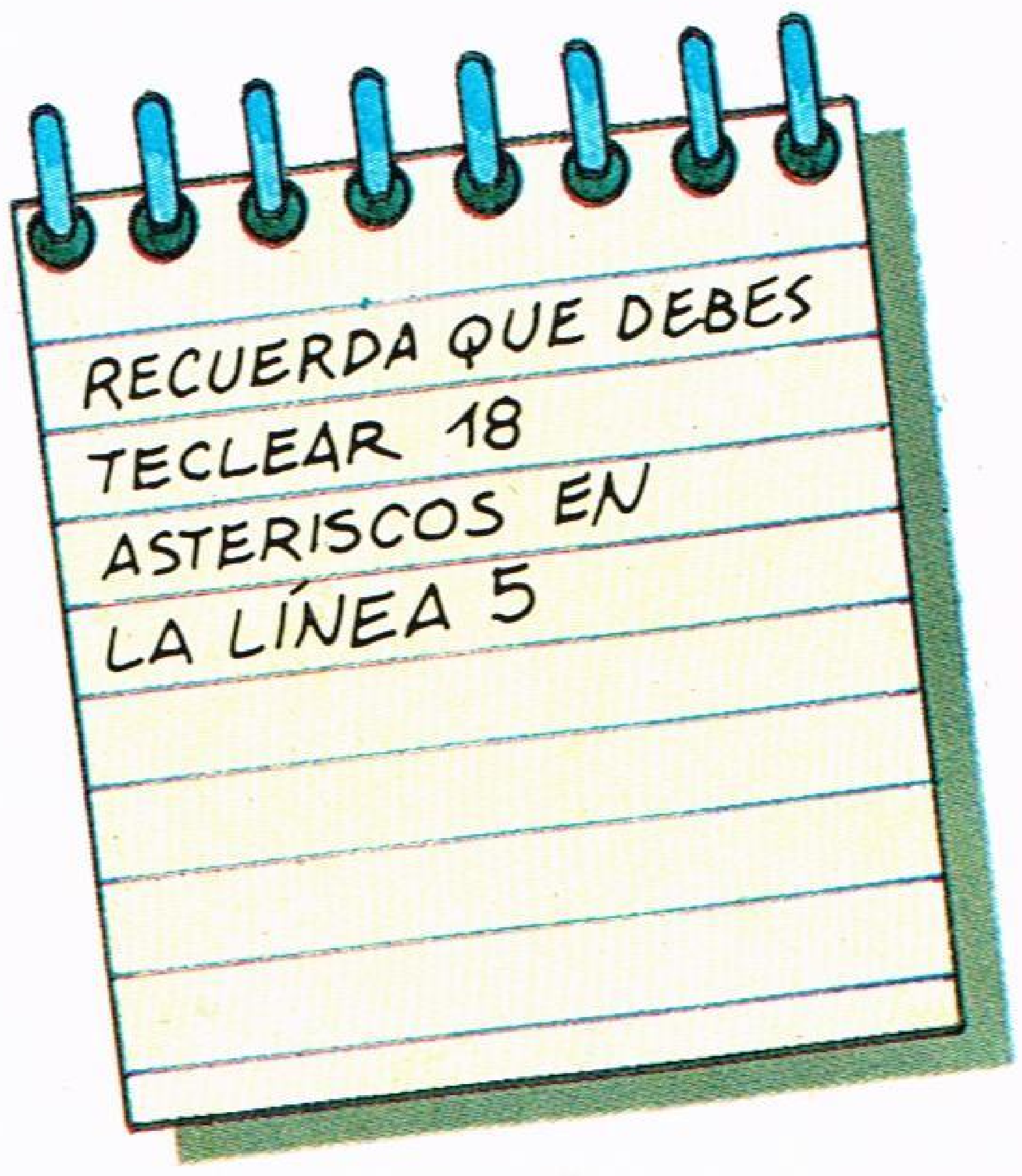
LOS ÁRBOLES SE DIBUJAN ACCEDIENDO DIRECTAMENTE A LA VRAM

Antes de continuar adelante graba el programa en cinta o disco, ya que un error en esta etapa puede bloquear por completo el ordenador.

La estructura del programa de control se ajusta al guión. La mayoría de las subrutinas se emplean una sola vez, como la usada para dibujar el castillo y los árboles. Las únicas subrutinas que se repiten son llamadas en las líneas 100 y 110 dentro del bucle **FOR... NEXT** constituido por las líneas 90 a 130. Estas subrutinas mueven el murciélago y la bruja unos pocos pixels cada vez que son ejecutadas.

```
  5 REM ******************
 10 REM casa encantada
 11 REM 18 bytes de codigo maquina almacenados en la
        linea 5
 20 GO SUB 6500: REM inicializar
 30 GO SUB 2000: REM fondo
 40 GO SUB 2500: REM castillo
 50 GO SUB 4500: REM arboles
 60 GO SUB 5000: REM oscuridad
 70 LET n=5: GO SUB 1500: REM rayo
 80 LET m=1300
 90 FOR b=0 TO 29
100 GO SUB 500: REM murcielago
110 GO SUB 1000: REM bruja
120 IF b=29 THEN  PRINT AT 1,29;"   ": REM esp,esp,esp
130 NEXT b
140 LET n=5: GO SUB 1500: REM rayo
150 GO SUB 5500: REM fuego
160 GO SUB 6000: REM calavera
170 STOP
```

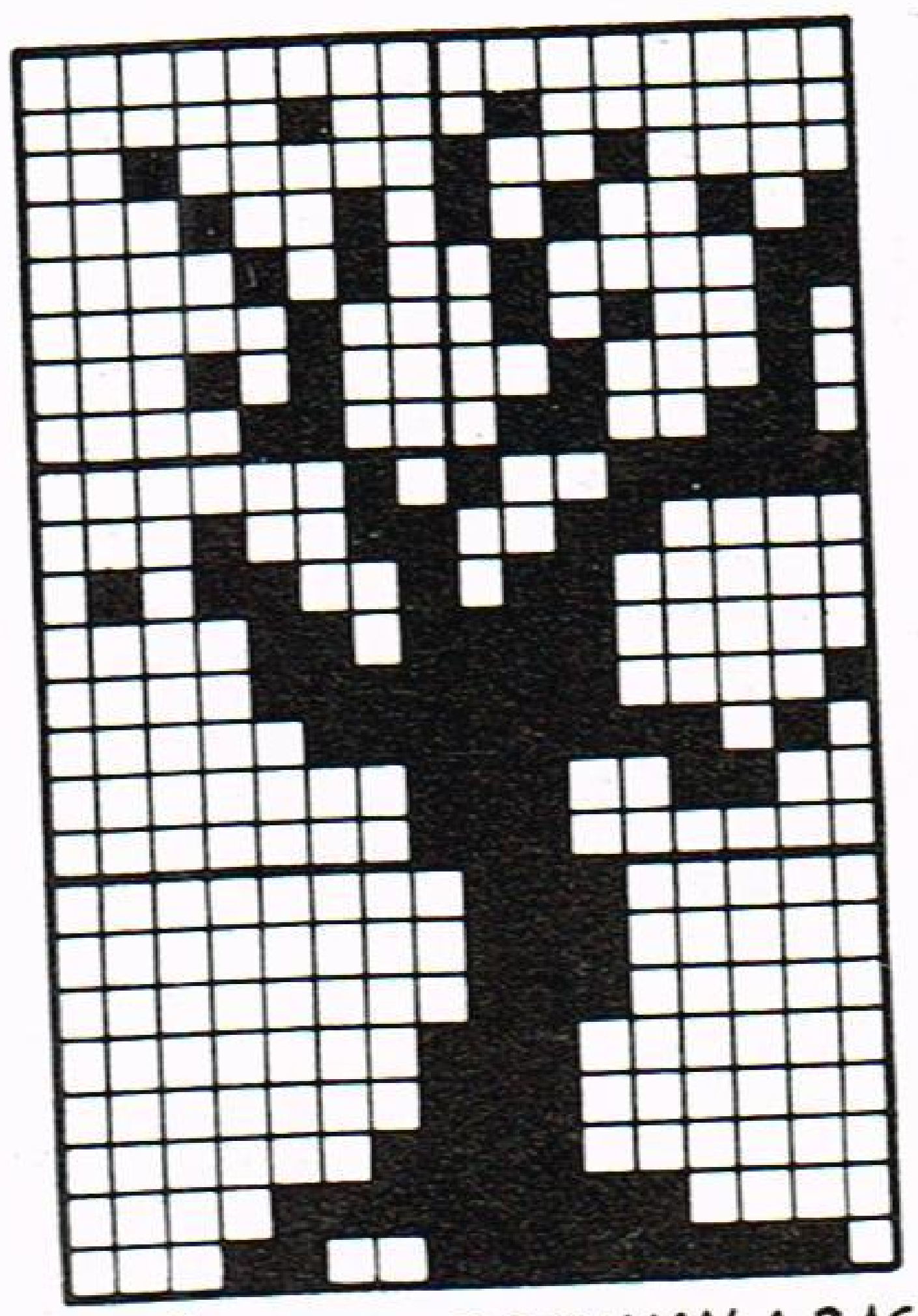
Como puedes observar en la línea 70 establecemos el valor de la variable **n** antes de entrar en la subrutina que dibuja el rayo. El valor de **n** nos dará el número de rayos a dibujar en esa subrutina (ver página 35). Es una buena técnica de programación escribir de esta forma las subrutinas que vamos a emplear bastante en el programa, ya que serán mucho más flexibles y versátiles. La línea 80 establece el valor de la variable **m** a su valor inicial antes de ser empleada en el bucle que mueve el murciélago y la bruja. Se ha preferido establecer su valor aquí en lugar de en la subrutina de inicialización para distinguir a qué parte del programa pertenece. La línea 120 se limita a imprimir tres espacios; éstos se emplean para borrar la bruja cuando esta alcanza la última posición de su trayectoria.



```
6500 REM inicializar
6510 FOR a=65368 TO 65527: READ d: POKE a,d: NEXT a
6520 REM introducir codigo maquina en la sentencia
     REM de la linea 5
6530 FOR i=23760 TO 23777: READ d: POKE i,d: NEXT i
6540 RETURN
```

```
7000 DATA 16,129,73,40,16,80,48,9: REM arbol (A)
7010 DATA 128,32,74,134,164,68,204,30: REM arbol (B)
7020 DATA 5,38,178,27,31,15,3,3: REM arbol (C)
7030 DATA 48,96,192,192,194,244,152,128: REM arbol (D)
7040 DATA 1,1,1,3,3,7,31,51: REM arbol (E)
7050 DATA 192,192,192,128,128,128,224,252: REM arbol(F)
7060 DATA 8,16,48,32,96,96,233,224: REM luna (G)
7070 DATA 240,254,127,113,48,54,31,6: REM luna (H)
```

La línea 6500 es el comienzo de la subrutina de inicialización. La línea 6510 lee los datos dados en las líneas 7000 a 7190 y los sitúa en las direcciones especiales reservadas para caracteres definidos por el usuario. Las letras entre paréntesis que hay detrás de cada línea de datos te indican en qué letra en modo gráfico se ha guardado cada carácter. La línea 6530 lee los datos de las líneas 7510 y 7520 y los emplea para crear una pequeña rutina en código máquina. Este programa nos permite cambiar instantáneamente determinados colores de la pantalla cuando se necesita cambiar la escena de día a noche. Una vez que hayas tecleado hasta la línea 7540 comprueba con mucho cuidado lo que acabas de escribir, ya que cualquier error en esta etapa puede bloquear el ordenador al intentar rodar el programa. Lo único que puedes hacer si esto ocurre es desconectarlo y comenzar de nuevo. De ahí que sea conveniente guardar el programa en cinta al finalizar esta etapa, de modo que simplemente debas cargarlo de nuevo en lugar de volverlo a teclear si algo ha fallado.

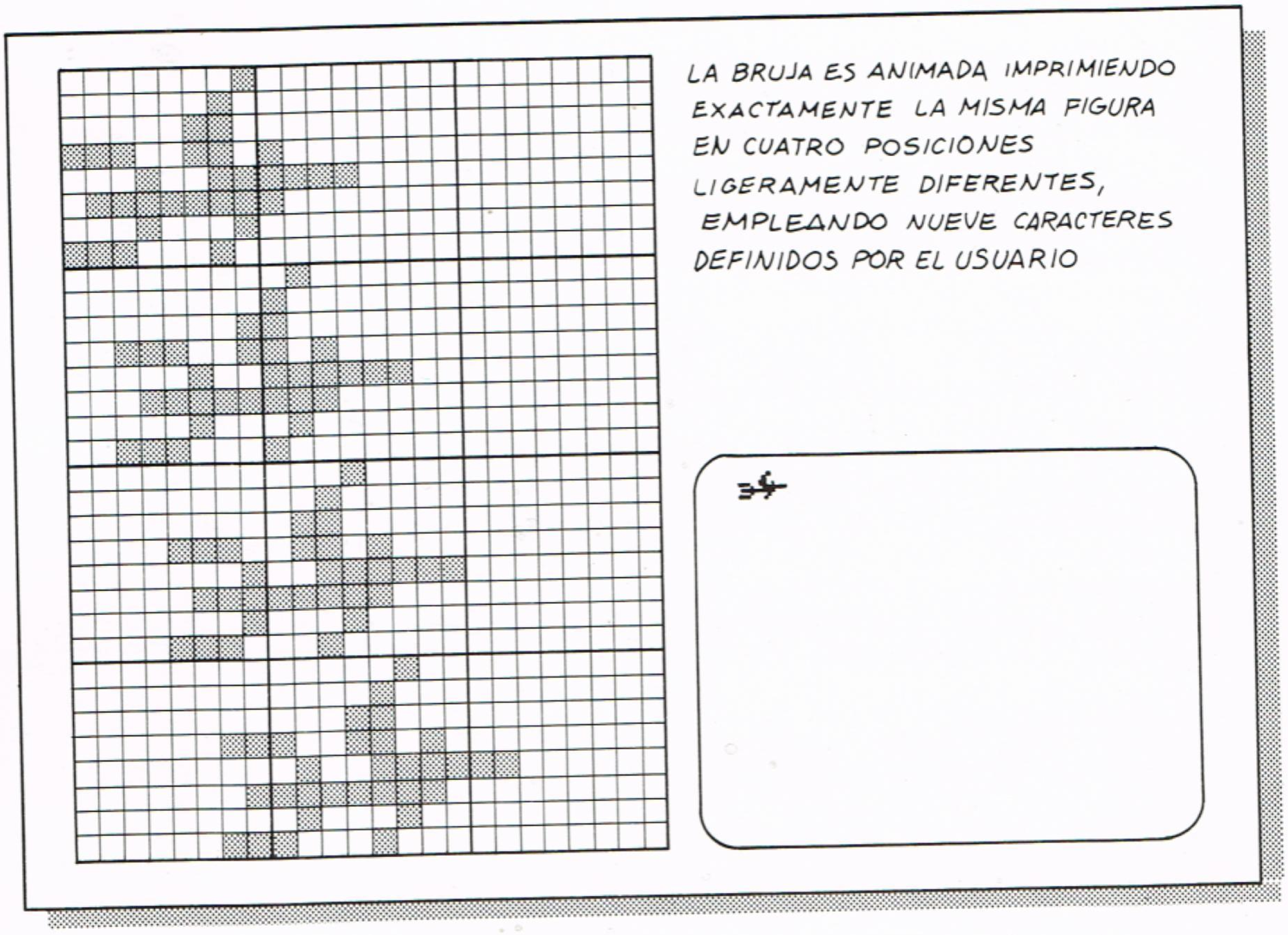


LOS ÁRBOLES SE DIBUJAN A BASE DE CARACTERES DEFINIDOS POR EL USUARIO

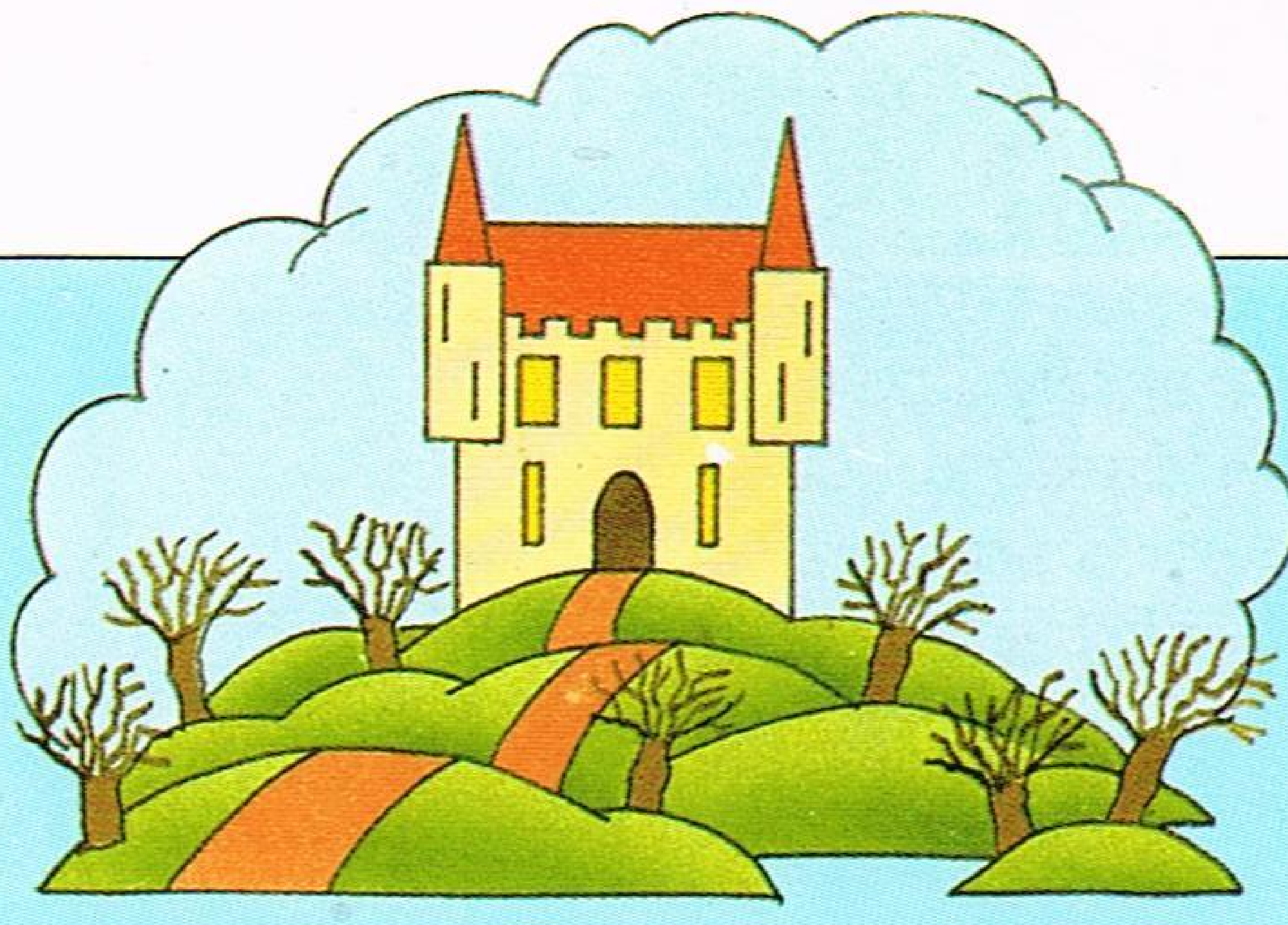
```
7080 DATA 1,2,6,230,19,255,17,226: REM bruja (I)
7090 DATA 0,0,0,128,240,128,0,0: REM bruja (J)
7100 DATA 0,0,1,57,4,31,4,56: REM bruja (K)
7110 DATA 64,128,128,160,252,224,64,128: REM bruja (L)
7120 DATA 0,0,0,14,1,7,1,14: REM  bruja (M)
7130 DATA 16,32,96,104,63,248,16,32: REM bruja (N)
7140 DATA 0,0,0,3,0,1,0,3: REM bruja (O)
7150 DATA 4,8,24,154,79,254,68,136: REM bruja (P)
7160 DATA 0,0,0,0,192,0,0,0: REM  bruja (Q)
7170 DATA 0,0,0,66,36,24,24,0: REM murcielago (R)
7180 DATA 0,0,0,0,0,255,24,0: REM murcielago (S)
7190 DATA 8,128,65,2,150,68,40,40: REM llamas (T)
7500 REM datos codigo maquina
7510 DATA 62,0,33,0,88,6,24,197,6,32
7520 DATA 119,35,16,252,193,16,246,201
7530 REM datos posicion ventanas
7540 DATA 10,22,8,23,14,19,14,12,10,9,8,8
```

Algunos de los caracteres del programa son creados empleando varias figuras; así, los árboles están constituidos por seis sentencias **DATA** que definen seis caracteres definidos por el usuario, como se muestra en la página anterior. La bruja, sin embargo, consta de nueve caracteres en total, ya que es animada empleando la misma técnica que la del avión en la página 10.

Puedes animar la forma que quieras empleando un programa similar y dibujando el gráfico en un diagrama como el que se muestra a continuación, imprimiendo las figuras en diferentes posiciones.

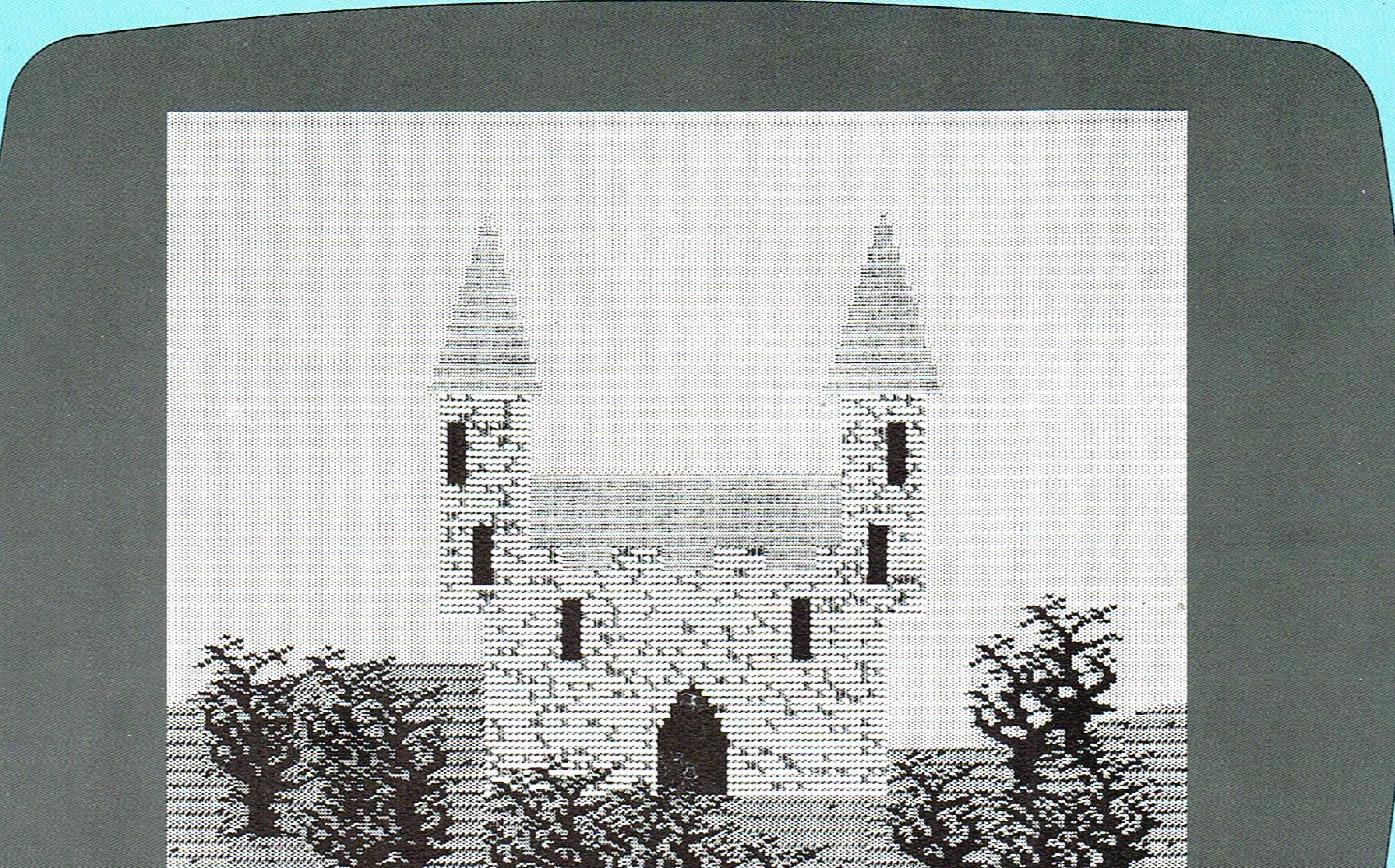


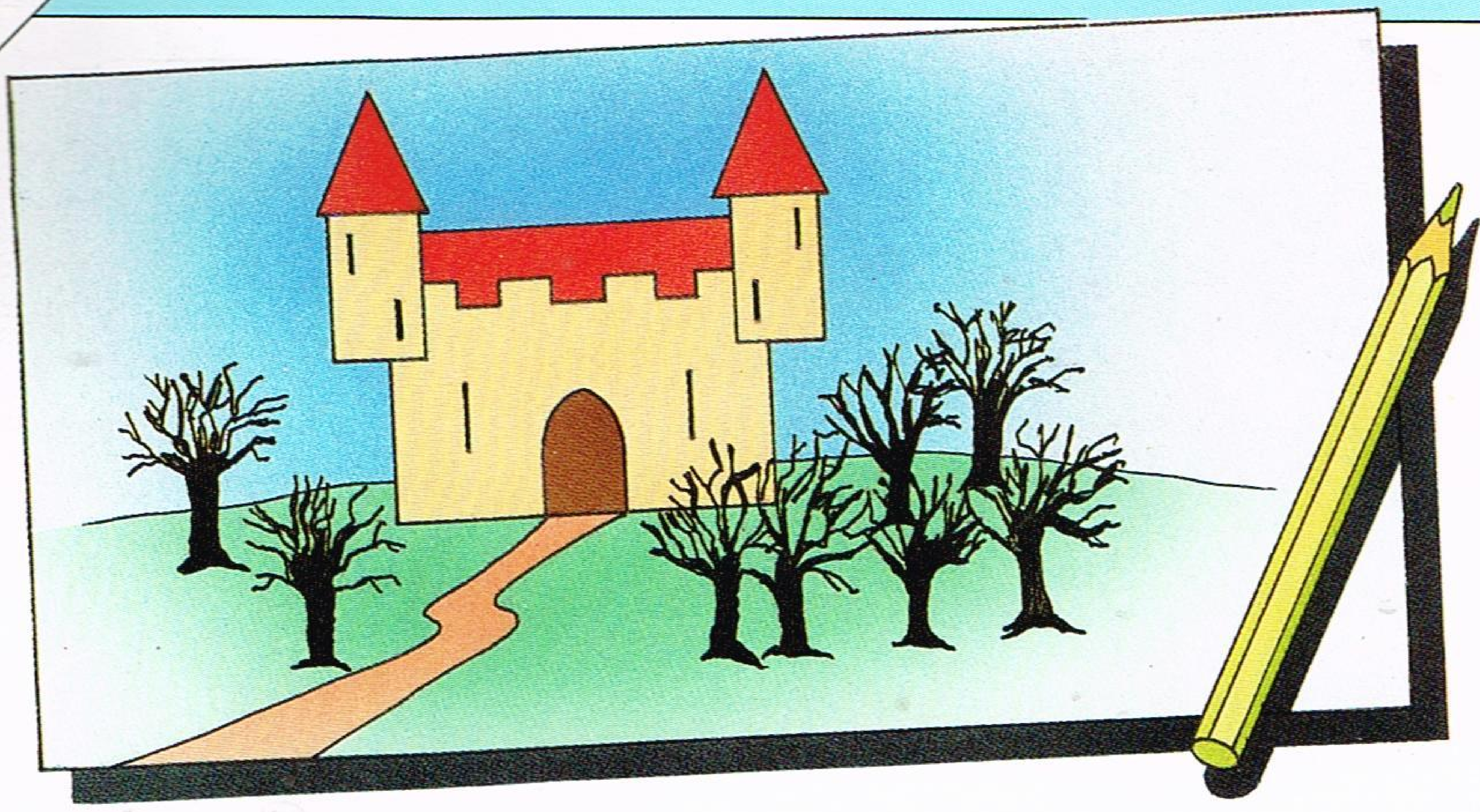
LA BRUJA ES ANIMADA IMPRIMIENDO EXACTAMENTE LA MISMA FIGURA EN CUATRO POSICIONES LIGERAMENTE DIFERENTES, EMPLEANDO NUEVE CARACTERES DEFINIDOS POR EL USUARIO

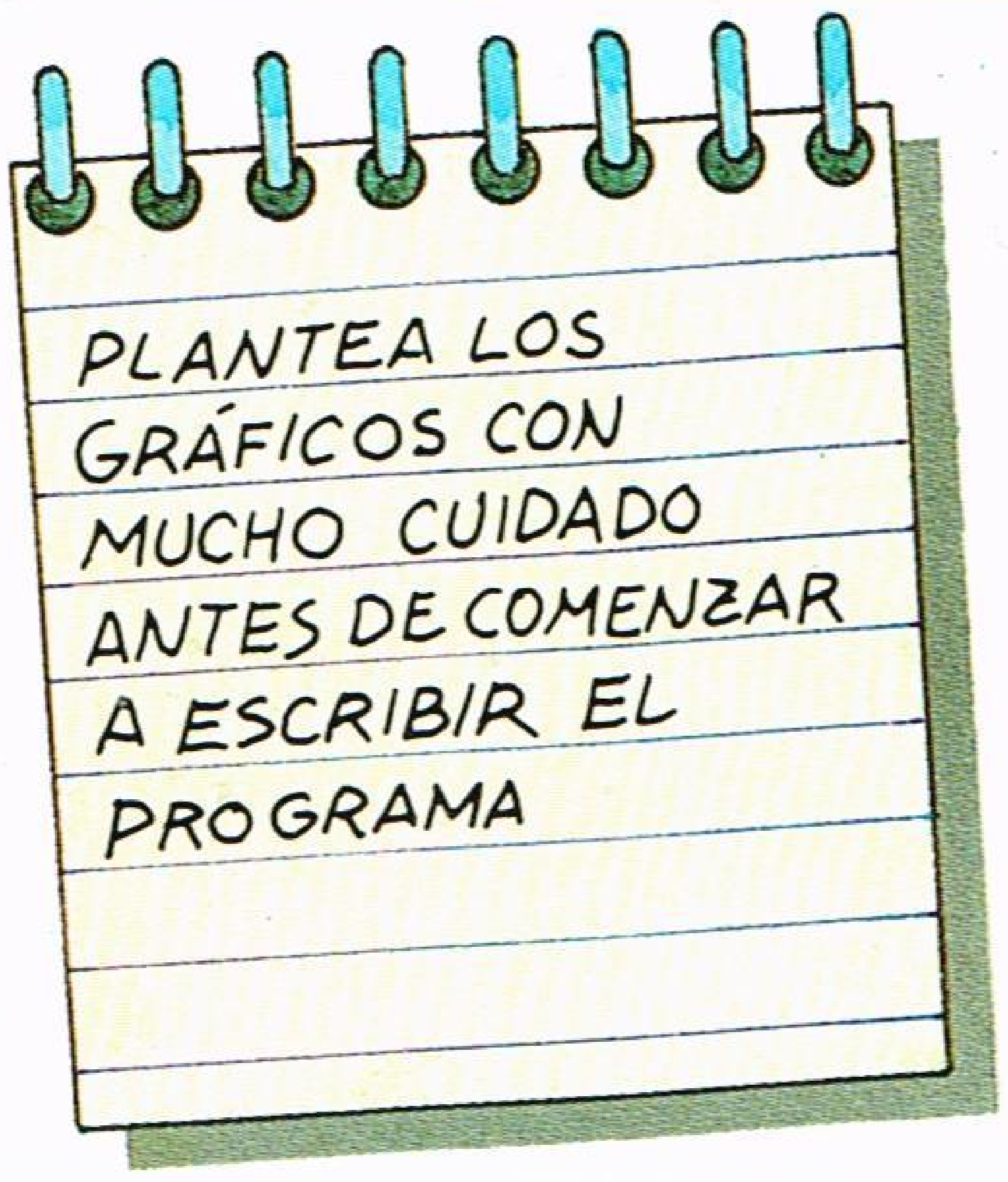


DIA

Esta sección del programa establece el escenario de fondo de día, la casa y el bosque fantasmagórico. También explica las subrutinas del programa de control incluida la sección que dibuja los árboles.







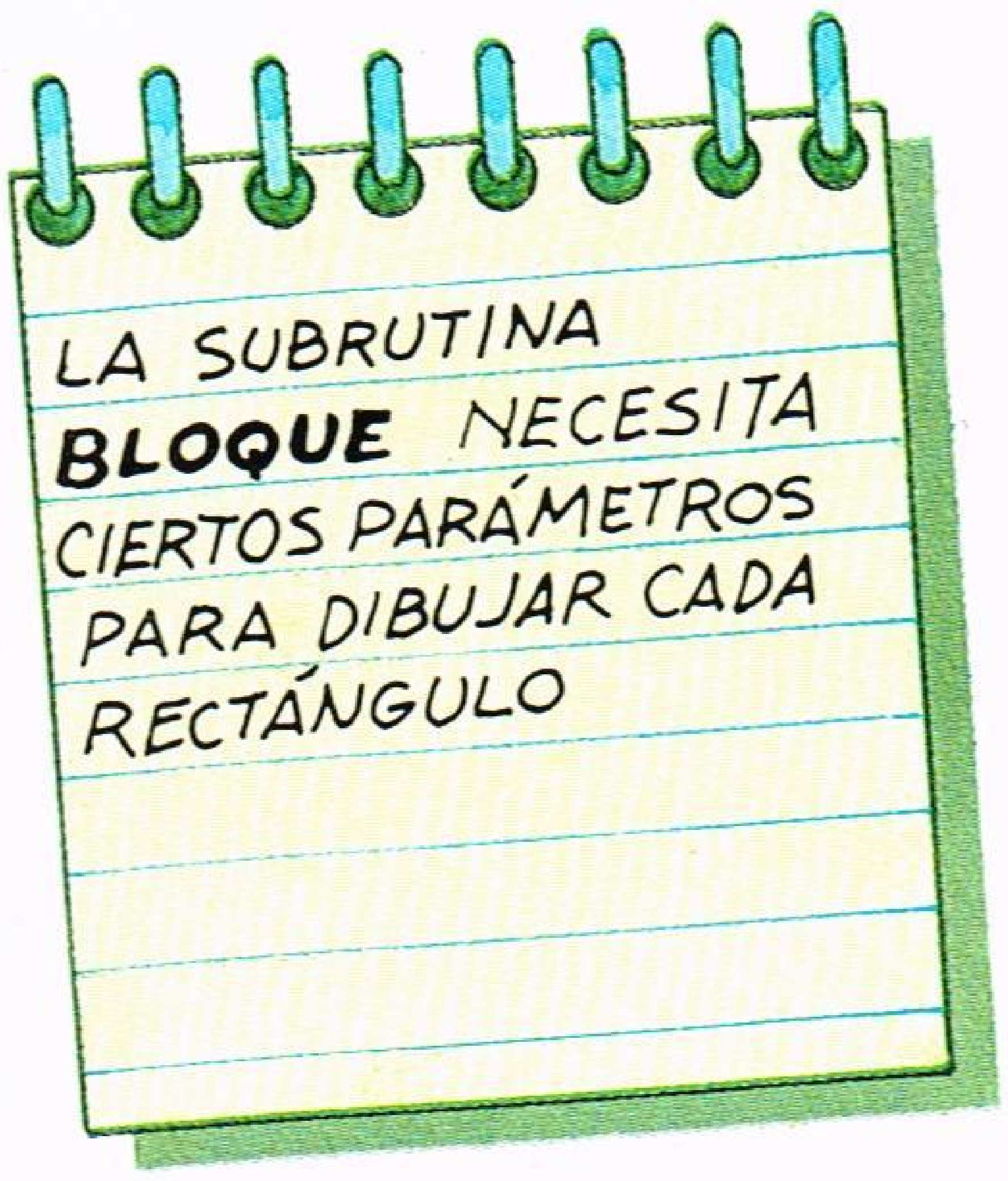
PLANTEA LOS
GRÁFICOS CON
MUCHO CUIDADO
ANTES DE COMENZAR
A ESCRIBIR EL
PROGRAMA

Esta subrutina dibuja el cielo y el primer plano, así como el camino que conduce a la puerta del castillo. La línea 10 del programa principal había establecido el color de fondo a verde y había borrado la pantalla. El bucle entre las líneas 2010 a 2030 dibuja 100 puntos negros situados en coordenadas aleatorias y simulan las piedras sobre el suelo. El cielo lo dibujaremos a base del carácter gráfico "■" (no empleamos el comando **PAINT** pues puede dar problemas al cambiar los colores con el programa en código máquina). Es necesario emplear el comando **OPEN "GRP:"** para imprimir dicho carácter, ya que estamos trabajando en modo 2 —modo gráfico—. Las líneas 2050 a 2070 son dos bucles anidados que se encargan de dibujar el cielo, comenzando desde la esquina superior izquierda y terminando en el límite impuesto a las piedras del suelo. Las líneas 2080 a 2110 dibujan el camino. Este se consigue trazando una serie de líneas horizontales cada vez más anchas que comienzan en puntos dados por una función seno.

```
2000 REM Fondo
2010 FOR R=1 TO 100
2020 X=INT (RND(1)*255):Y=130+INT (RND(1)*60)
2030 PSET (X,Y),1:NEXT
2040 OPEN "GRP:" AS #1:COLOR 5
2050 FOR Y=0 TO 120 STEP 8:FOR X=0 TO 248 STEP 8
2060 PRESET (X,Y):PRINT#1,"■"
2070 NEXT X:NEXT Y
2080 I=16
2090 FOR P=63 TO 0 STEP -2
2100 X=138*SIN(P/60):Y=192-P:LINE (X,Y)-(X+I,Y),1
2110 I=I+1.5:NEXT
2120 RETURN
```

```
2500 REM Castillo
2510 X1=80:Y1=72:X2=168:Y2=120:LAD=300:
     GOSUB 3000:REM Bloque frontal
2520 X1=64:Y1=40:X2=80:Y2=80:LAD=70:
     GOSUB 3000:REM Torre izquierda
2530 X1=168:Y1=40:X2=184:Y2=80:LAD=70:
     GOSUB 3000:REM Torre derecha
2540 XS=76:YS=8:GOSUB 3500:REM Tejado izquierdo
2550 XS=180:YS=8:GOSUB 3500:REM Tejado derecho
2560 FOR Y=60 TO 72
2570 LINE (89,Y)-(167,Y),6
2580 NEXT
```

El castillo lo dibujamos mediante la subrutina que comienza en la línea 2500. Consta de tres volúmenes principales que se consiguen mediante la subrutina **Bloque** que explicaremos posteriormente. Antes de entrar en la misma es necesario establecer los parámetros que le servirán para determinar los límites de cada bloque. Las líneas 2510 a 2530 definen la posición de cada volumen así como el valor de la variable **LAD** (que emplearemos para simular el enladrillado) y llaman a dicha subrutina. A continuación las líneas 2540 y 2550 actúan de forma parecida sobre la subrutina **Segmento** encargada de dibujar los tejados de las dos torres laterales; establecen el valor del vértice de cada una y llaman a la subrutina.
El bucle **FOR... NEST** entre las líneas 2560 y 2580 dibuja el tejado principal mediante unas líneas horizontales entre las dos torres. Para dibujar las seis ventanas emplearemos también otra subrutina —que comienza en la línea 4000— y que, como las anteriores, necesita de dos parámetros que indiquen la posición de cada ventana. Por fin la línea 2660 dibuja la puerta a base de líneas negras de anchura creciente hasta que alcanzan un máximo y lo mantienen.

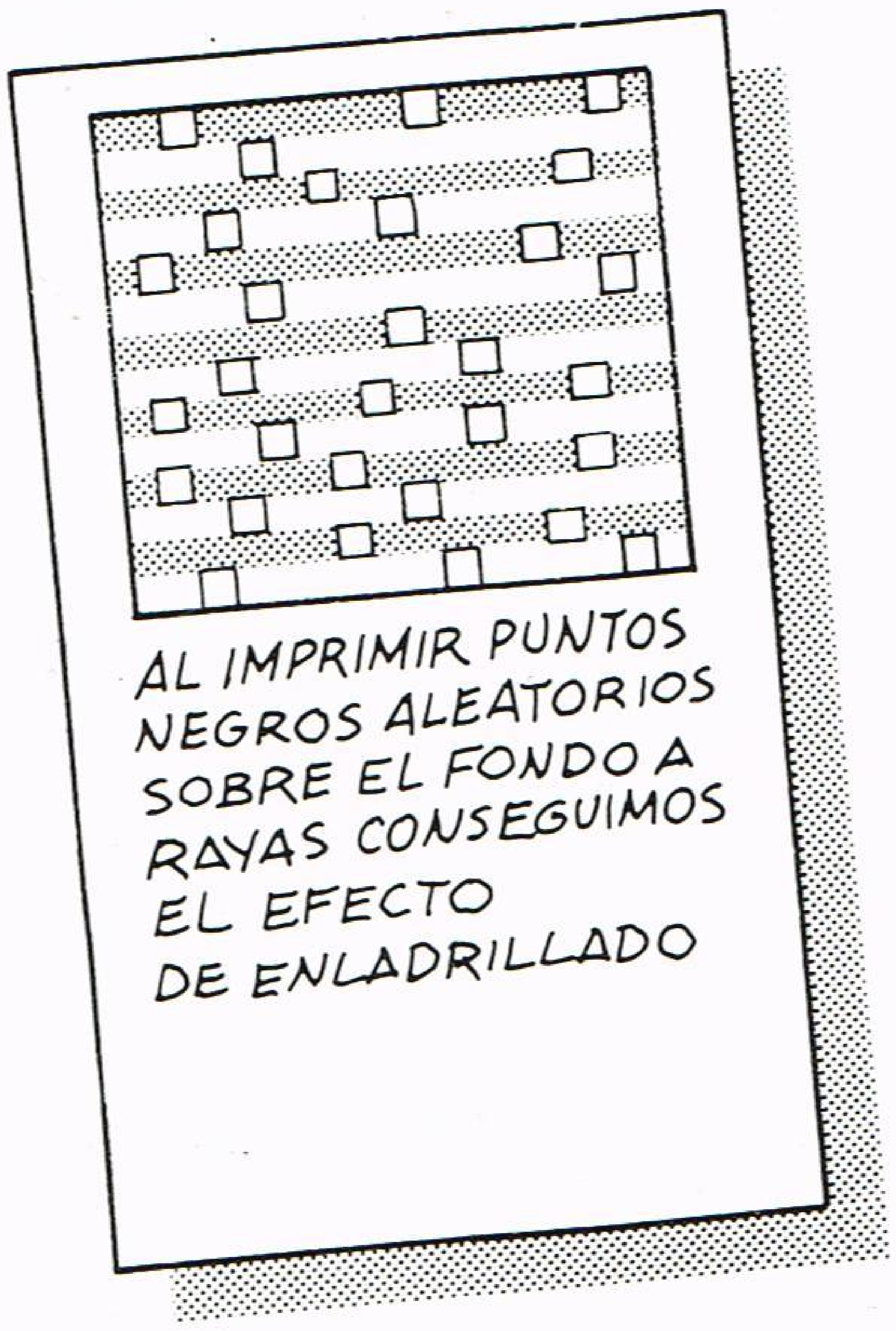


LA SUBRUTINA **BLOQUE** NECESITA CIERTOS PARÁMETROS PARA DIBUJAR CADA RECTÁNGULO

```
2600 X=68:Y=47:GOSUB 4000
2610 X=76:Y=63:GOSUB 4000
2620 X=182:Y=47:GOSUB 4000
2630 X=174:Y=63:GOSUB 4000
2640 X=96:Y=83:GOSUB 4000
2650 X=154:Y=83:GOSUB 4000
2660 FOR Y=104 TO 127:IF Y<112 THEN
     LINE (232-Y,Y)-(24+Y,Y),1:NEXT
     ELSE LINE (120,Y)-(136,Y),1:NEXT
2670 RETURN
```

```
3000 REM Bloque
3010 FOR Y=Y1 TO Y2 STEP 8:FOR X=X1 TO X2 STEP 8
3020 COLOR 9:PRESET (X,Y):PRINT #1,"█"
3030 NEXT X:NEXT Y
3040 FOR Y=Y1 TO Y2+7 STEP 2
3050 LINE (X1,Y)-(X2+7,Y),1
3060 NEXT Y
3070 COLOR 1:FOR M=1 TO LAD
3080 A=INT (RND(1)*(X2-X1+7)+X1):
     B=INT (RND(1)*(Y2-Y1+7)+Y1):PSET (A,B),1
3090 NEXT M
3100 RETURN
```

La subrutina **Bloque** es la encargada de dibujar el frente del castillo y las dos torres. El efecto de enladrillado se consigue en tres etapas. En primer lugar se dibuja un fondo rojo. Para ello es necesario conocer los límites del rectángulo a dibujar; éstos vienen dados por **X1**, **X2**, **Y1** e **Y2** y son actualizados antes de entrar en la subrutina. Los bucles anidados entre las líneas 3010 y 3030 se encargan de dibujar el rectángulo base. La segunda etapa consiste en dibujar una serie de líneas negras horizontales sobre el rectángulo —líneas 3040 a 3060—; hasta ahora lo que tenemos es un rectángulo rayado en rojo y negro. Para terminar el efecto de enladrillado dibujaremos una serie de puntos negros sobre el bloque —líneas 3070 a 3090—. Las coordenadas de estos puntos no deben rebasar los límites del muro. Los puntos negros sólo se verán cuando estén sobre rayas rojas.



AL IMPRIMIR PUNTOS NEGROS ALEATORIOS SOBRE EL FONDO A RAYAS CONSEGUIMOS EL EFECTO DE ENLADRILLADO

La subrutina **Segmento** —línea 3500— dibuja una serie de radios de circunferencia de centro **XS**, **YS** —parámetros asignados antes de entrar en la subrutina— y simula el tejado de las torres laterales. Por fin las ventanas son dibujadas mediante la subrutina que comienza en la línea 4000 por un procedimiento análogo al de la puerta.

```
3500 REM Segmento
3510 R=38
3520 FOR T=160 TO 200 STEP 2
3530 A=T*3.14/180:X=R*SIN(A):Y=R*COS(A)
3540 LINE (XS,YS)-(XS+X,YS-Y),6
3550 NEXT
3560 RETURN
```

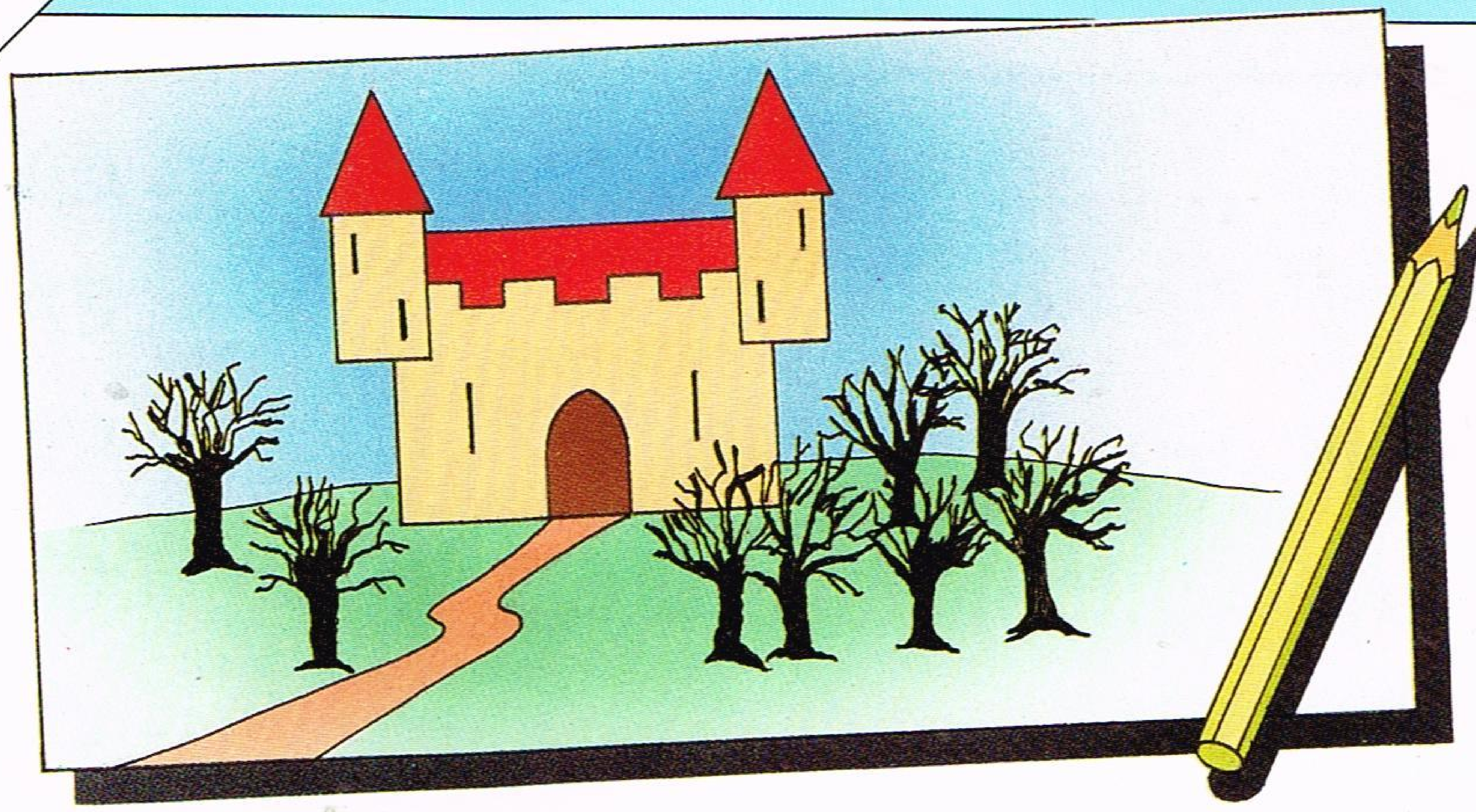
```
4000 REM Ventana
4010 I=5
4020 FOR J=Y TO Y+17:IF J<Y+5 THEN
     LINE (X+I/2,J)-(X+6-I/2,J),1:I=I-1:NEXT
     ELSE LINE (X,J)-(X+6,J),1:NEXT
4030 RETURN
```

```
4500 REM Arboles
4510 DIM A(21)
4520 FOR J=1 TO 21:READ N:A(J)=N:NEXT
4530 FOR J=1 TO 21
4540 I=BASE(10)+A(J)
4550 VPOKE (I),0:VPOKE (I+1),1:VPOKE (I+32),32:
     VPOKE (I+33),33:VPOKE (I+64),64:VPOKE (I+65),65
4560 NEXT J
4570 R=1000:GOSUB 300
4580 RETURN
```

Sólo queda añadir los árboles para completar la escena. Esto lo lleva a cabo la subrutina que comienza en la línea 4500. El proceso ya se ha indicado en la introducción. En primer lugar hay que definir la forma de los árboles accediendo a la tabla de patrones (ésto ya lo hicimos en la subrutina de inicialización, líneas 6540 a 6560). A continuación hay que indicar en la tabla de nombres, en cada posición de pantalla que queremos que contenga parte del árbol, dónde se encuentra la información que define dicha porción. Una vez hecho esto aparecerá el árbol en pantalla.

En primer lugar la línea 4510 **DIM**ensiona la matriz que va a contener las posiciones de pantalla correspondientes a la esquina superior izquierda de cada árbol. El bucle 4520 asigna a esta matriz los valores correspondientes. A continuación el lazo constituido por las líneas 4530 a 4560 se encarga de alterar las posiciones oportunas de la tabla de nombres para indicar dónde se encuentra la definición de cada porción de árbol. La línea 4570 introduce un retardo que lleva a cabo la subrutina que comienza en la línea 300 —un simple bucle **FOR... NEXT** que cuenta de 1 a **R**.

Con esta subrutina has completado la parte del programa encargada de dibujar el escenario de la acción.



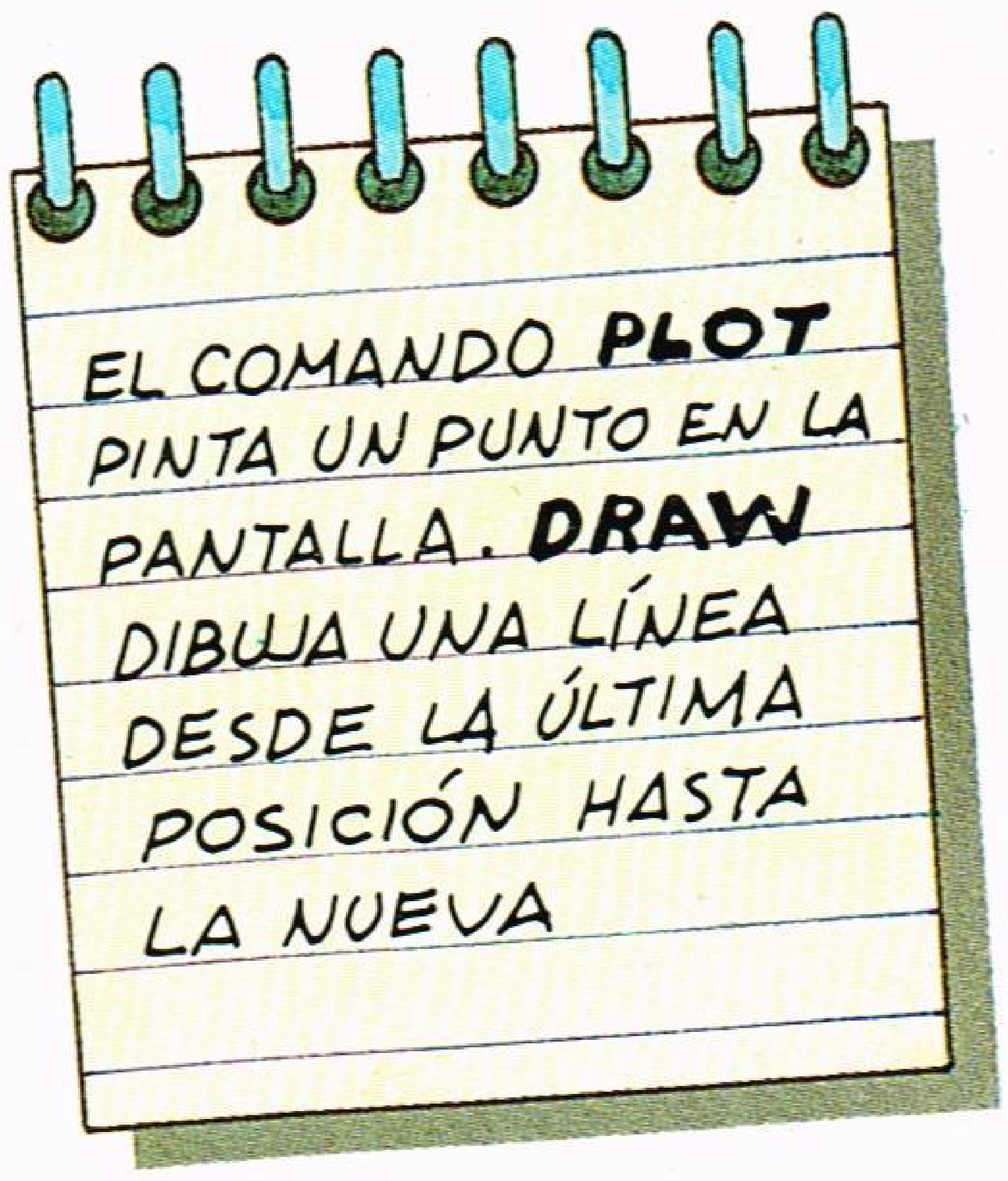
EL CAMINO SE
CONSIGUE MEDIANTE
UNA FUNCIÓN
SENO.

Debes tener mucho cuidado al proyectar cualquier clase de gráficos en el Spectrum, ya que éste no te permite usar mas que un color para el fondo y otro para la tinta en cada carácter. La subrutina **Fondo,** que comienza en la línea 2000 borra la pantalla y la pasa a color verde **(PAPER 4)**. Un bucle **FOR...NEXT** elige 100 coordenadas aleatorias en la mitad inferior de la pantalla y dibuja un punto negro en cada una. Estos puntos simulan rocas, y ayudan a romper la monotonía del suelo. El cielo se dibuja cambiando el color a cyan **(PAPER 5)** y usando dos bucles **FOR... NEXT** anidados para imprimir una serie de espacios coloreados en la mitad superior de la pantalla. La línea 2080 pasa la tinta a color negro para dibujar la línea del horizonte que separará tierra y cielo. A continuación se fija el valor de la variable **i** a cuatro antes de entrar en otro bucle que se encarga de dibujar el camino. Este lo conseguiremos trazando una serie de líneas horizontales cada vez más anchas que comienzan en puntos dados por una función seno. La anchura creciente se consigue incrementando el valor de la variable **i** que nos da el número de pixels que debe contener cada línea horizontal. La subrutina finaliza en la línea 2130.

```
2000 REM fondo
2010 PAPER 4: INK 0: CLS
2020 FOR r=1 TO 100
2030 LET x=INT (RND*255): LET y=INT (RND*80): PLOT x,y
2040 NEXT r: PAPER 5
2050 FOR y=0 TO 12: FOR x=0 TO 31
2060 PRINT AT y,x;" ": REM espacio
2070 NEXT x: NEXT y
2080 INK 0
2090 PLOT 0,72: DRAW 255,0: LET i=4
2100 FOR p=39 TO 0 STEP -2
2110 PLOT 130*SIN (p/30),p: DRAW i,0: LET i=i+2
2120 NEXT p
2130 RETURN
```

```
2500 REM castillo
2510 LET x1=10: LET x2=21: LET y1=10: LET y2=16:
     LET lad=400: GO SUB 3000: REM bloque frontal
2520 LET x1=8: LET x2=10: LET y1=6: LET y2=11:
     LET lad=100: GO SUB 3000: REM bloque torre izda.
2530 LET x1=21: LET x2=23: LET y1=6: LET y2=11:
     LET lad=100: GO SUB 3000: REM bloque torre dcha.
2540 LET xs=75: LET ys=159:
     GO SUB 3500: REM segmento tejado torre izda.
2550 LET xs=179: LET ys=159:
     GO SUB 3500: REM segmento tejado torre dcha.
2560 FOR r=88 TO 103
2570 PLOT 88,r: DRAW 79,0
2580 NEXT r
```

La subrutina **castillo**, que comienza en la línea 2500, se encarga de dibujar el edificio. No obstante es la subrutina **bloque** —línea 3000— la que consigue dibujar el enladrillado del frente y las dos torres laterales. Las variables **x1**, **x2**, **y1**, e **y2** establecen la posición del bloque que debe ser rellenado de ladrillos, y se estudiarán con detalle en la página siguiente al describir la subrutina **bloque.** Las líneas 2540 y 2550 establecen los vértices de los tejados de las torres antes de llamar a la subrutina 3500 que se encarga de dibujarlos. Un bucle **FOR.... NEXT** entre las líneas 2560 y 2580 dibuja el tejado principal mediante una serie de líneas horizontales entre las dos torres. La línea 2590 imprime una serie de cuadrados para representar las almenas de la parte frontal. El símbolo 8 entre las comillas indica que debes poner el ordenador en **MODO GRAFICO** (pulsa CAPS SHIFT y 9) y pulsar la tecla 8. A continuación se dibujan seis ventanas mediante la subrutina 4000. Como antes las líneas 2600 a 2650 definen la posición de cada ventana antes de entrar en la subrutina que se encarga de dibujarlas. Por último la línea 2660 emplea el carácter gráfico 8 para imprimir una puerta. Esta consta de cuatro caracteres de este tipo.

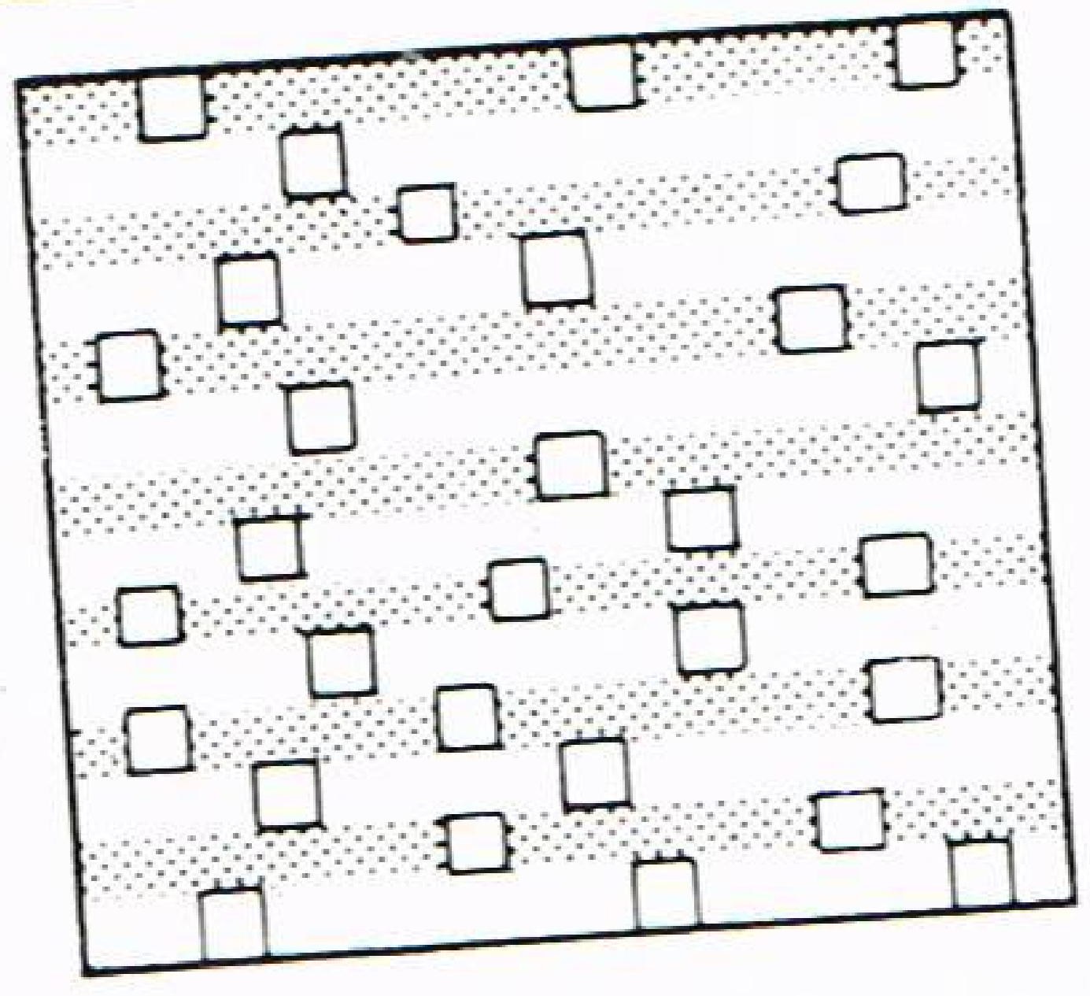


EL COMANDO **PLOT** PINTA UN PUNTO EN LA PANTALLA. **DRAW** DIBUJA UNA LÍNEA DESDE LA ÚLTIMA POSICIÓN HASTA LA NUEVA

```
2590 FOR b=12 TO 20 STEP 2:
     PRINT AT 11,b;"8": NEXT b: REM (8)
2600 LET x=68: LET y=102: GO SUB 4000
2610 LET x=76: LET y=88: GO SUB 4000
2620 LET x=182: LET y=102: GO SUB 4000
2630 LET x=174: LET y=88: GO SUB 4000
2640 LET x=96: LET y=56: GO SUB 4000
2650 LET x=157: LET y=56: GO SUB 4000
2660 PRINT AT 15,15;"88";AT 16,15;"88":
     REM (8),(8);(8),(8)
2670 RETURN
```

```
3000 REM bloque
3010 FOR y=y2 TO y1 STEP -1: FOR x=x1 TO x2
3020 PAPER 2: PRINT AT y,x;" ": REM espacio
3030 NEXT x: NEXT y
3040 LET x1=x1*8: LET x2=x2*8: LET y1=y1*8: LET y2=y2*8
3050 FOR y=y2 TO y1 STEP -2
3060 INK 0: PLOT x1,168-y: DRAW (x2-x1)+7,0
3070 NEXT y
3080 FOR m=1 TO lad
3090 LET a=INT (RND*(x2-x1+7)+x1):
     LET b=168-INT (RND*(y2-y1)+y1): PLOT a,b
3100 NEXT m
3110 RETURN
```

La subrutina **bloque** se emplea para dibujar el frente de la casa y las dos torres laterales. Veamos cómo conseguir el efecto de un muro de ladrillos. En primer lugar se dibuja una serie de espacios en rojo **(PAPER 2)** entre los límites dados por las coordenadas mediante los bucles **FOR... NEXT** en las líneas 3010 a 3030. De esta forma conseguimos la base del muro. Sobre ésta se dibuja una serie de líneas negras horizontales comenzando en pixels alternados a lo largo del eje y mediante el bucle entre las líneas 3050 y 3070. Hasta ahora lo que tenemos es un rectángulo rayado en rojo y negro. Para transformarlo en un muro enladrillado dibujaremos una serie de puntos negros situados al azar en dicho rectángulo. La variable **lad** contiene el número de puntos a distribuir. **PLOT a,b**, se encarga de dibujar dichos puntos en la posición x,y. Estas coordenadas se calculan de modo que se encuentren siempre dentro de los límites del muro. Los puntos negros sólo se verán cuando se dibujen sobre líneas rojas. De esta forma se completa el efecto de enladrillado. La subrutina **segmento** —que comienza en la línea 3500— se limita a dibujar un sector de círculo de radio 39 pixels y de centro las coordenadas **xs, ys** para conseguir los tejados de las dos torres laterales. Las ventanas se dibujan en la subrutina que comienza en la línea 4000 mediante un lazo que traza una serie de pequeñas líneas negras horizontales.



EL MURO ENLADRILLADO SE CONSIGUE DISTRIBUYENDO PUNTOS NEGROS ALEATORIOS SOBRE LOS TRES RECTÁNGULOS EN ROJO Y NEGRO QUE CONSTITUYEN LA CASA

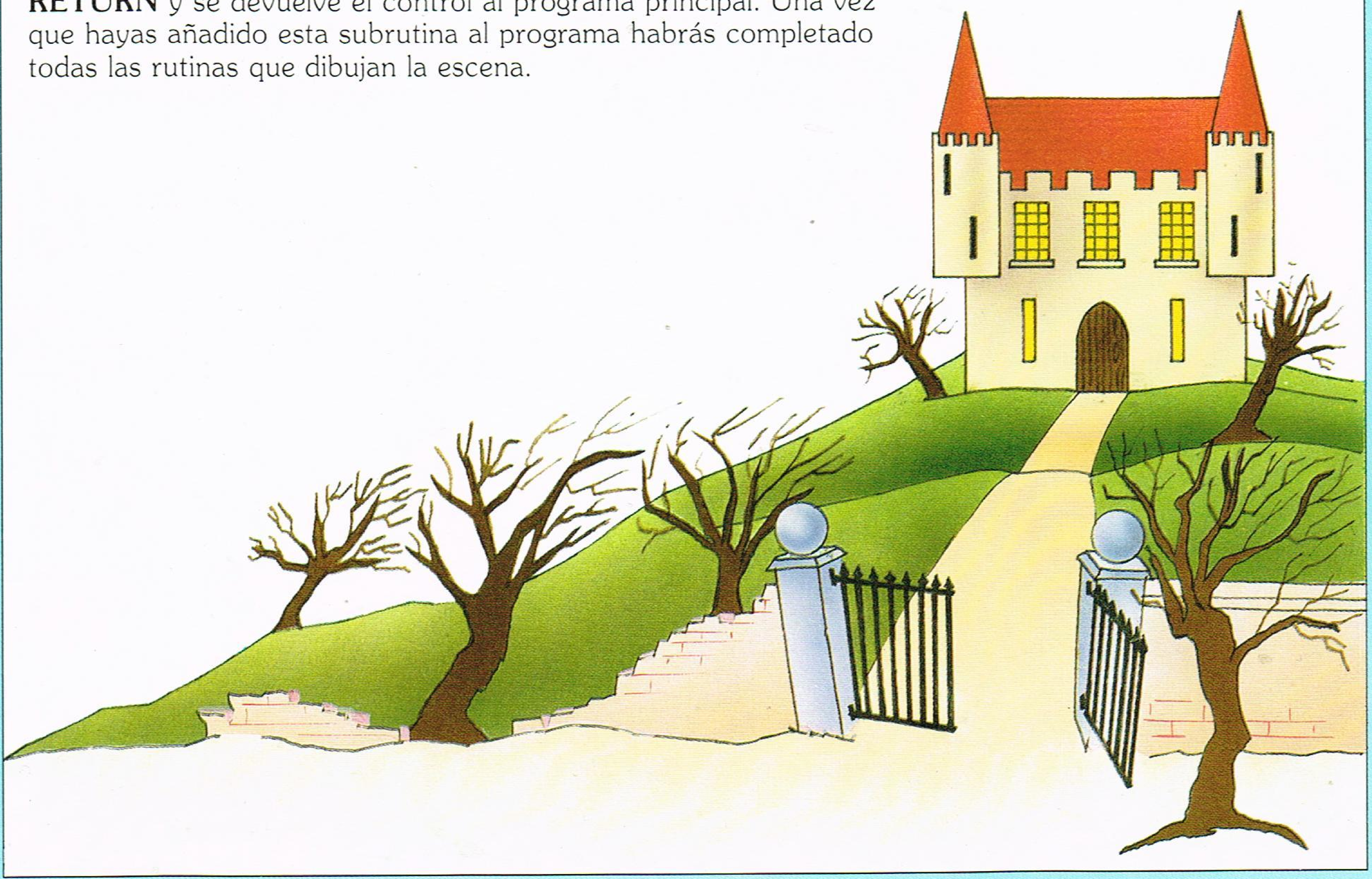
```
3500 REM segmento
3510 LET r=39: INK 3
3520 FOR t=160 TO 200
3530 LET a=t*PI/180: LET x=r*SIN (a): LET y=r*COS (a)
3540 PLOT xs,ys: DRAW x,y
3550 NEXT t
3560 RETURN
4000 REM ventana
4010 INK 0: FOR w=y TO y+10: PLOT x,w: DRAW 3,0
4011 NEXT w
4020 RETURN
```

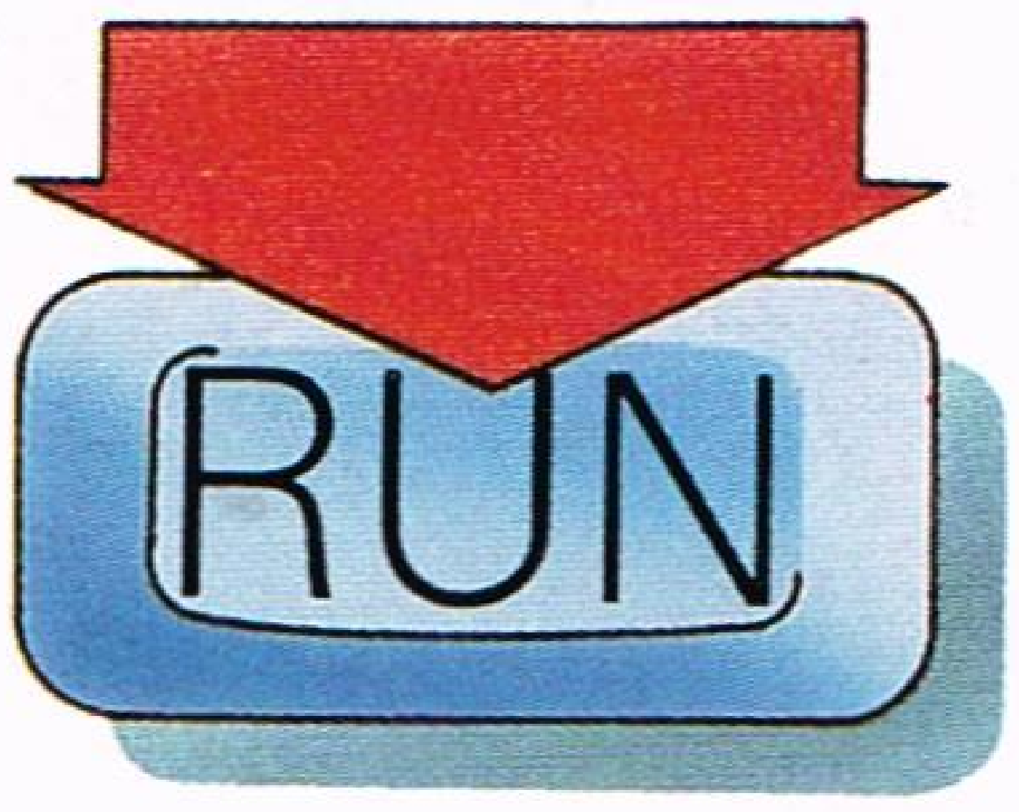
```
4500 REM arboles
4510 PAPER 4: INK 0: OVER 1
4520 FOR t=1 TO 30
4530 LET x=INT (RND*29): LET y=13+INT (RND*7)
4540 IF (x>8 AND x<22 AND y<17) THEN  GO TO 4530
4550 PRINT AT y,x;"AB";AT y+1,x;"CD";AT y+2,x;"EF":
     REM (A),(B);(C),(D);(E),(F)
4560 NEXT t
4570 OVER 0: PAUSE 200
4580 RETURN
```

Sólo queda añadir el bosque siniestro para completar la escena de día. La subrutina de la línea 4500 dibuja 30 árboles en posiciones aleatorias sobre la hierba. Cada árbol tiene un color verde de fondo **(PAPER 4)**; de ahí que sea necesario asegurarse de que no se va a imprimir en el cielo —línea 4530— ni sobre la casa —línea 4540.

Cada árbol se compone de seis caracteres, agrupados de dos en dos, en tres bloques (línea 4550). La instrucción **OVER 1** en la línea 4510 tiene el efecto de hacer un carácter transparente, de modo que al imprimirlo sobre otro se vean ambos. Cuando parte de un árbol se imprime sobre otro es posible ver a través del que está en primer plano. Una vez que se han dibujado todos los árboles, la instrucción **OVER 0** en la línea 4570 recobra la impresión normal.

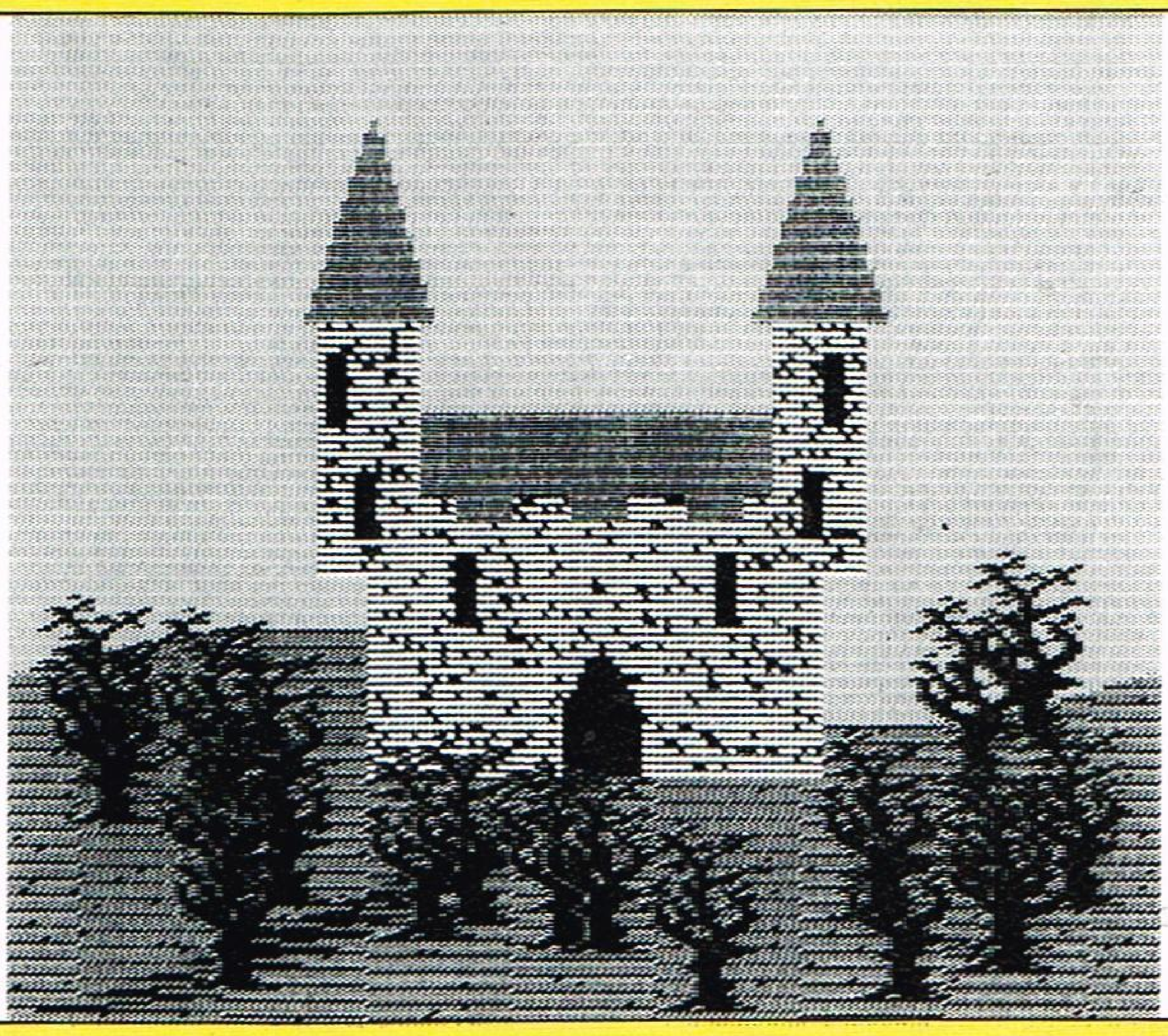
Tras una pequeña pausa la subrutina finaliza con una instrucción **RETURN** y se devuelve el control al programa principal. Una vez que hayas añadido esta subrutina al programa habrás completado todas las rutinas que dibujan la escena.



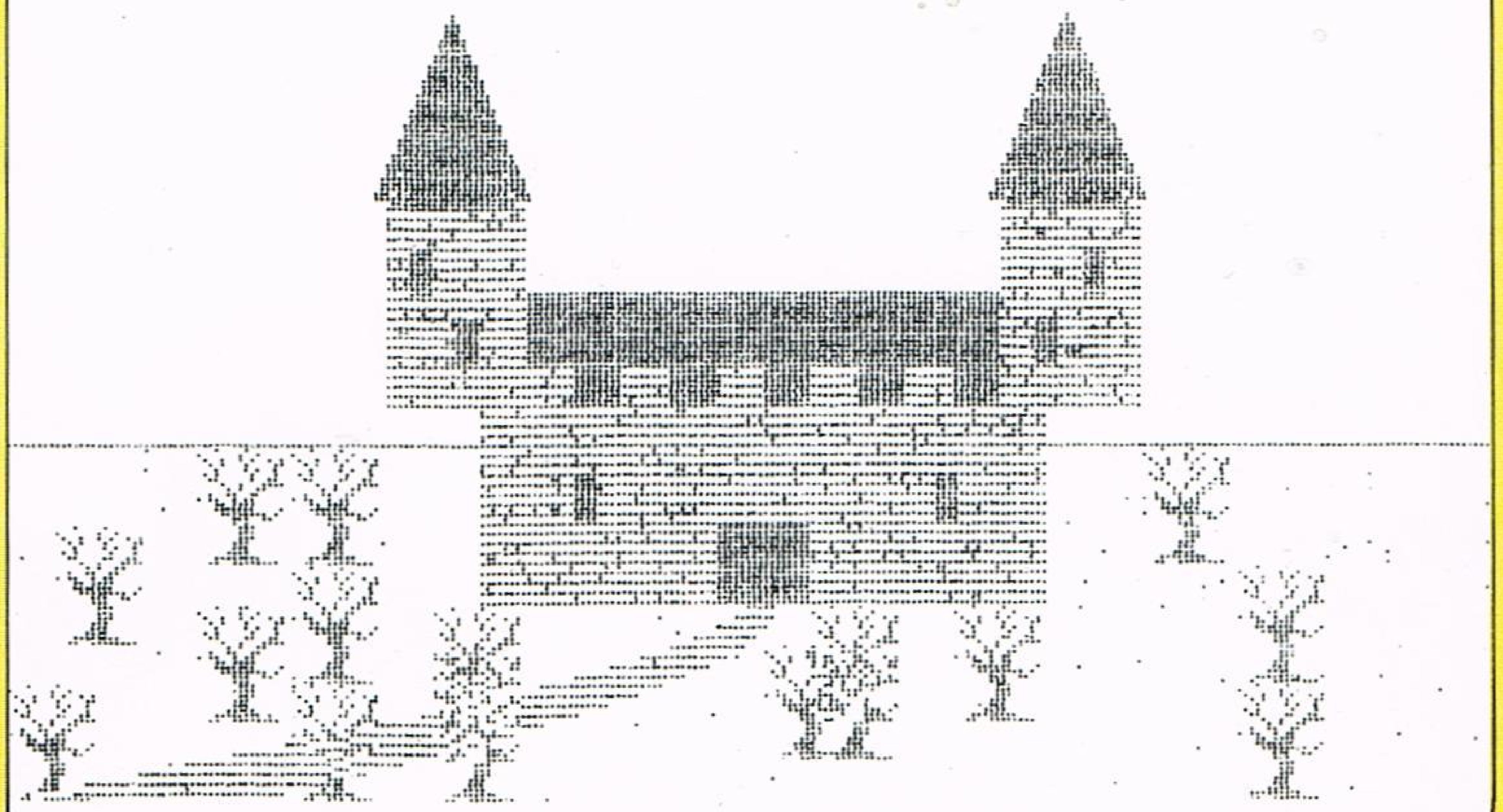


Prueba tu programa

Antes de continuar introduciendo la siguiente parte del programa conviene guardarlo en cinta ("salvarlo"). Encontrarás las instrucciones oportunas en el manual del usuario. Puedes, no obstante, probar lo que llevas hecho hasta ahora siguiendo las instrucciones de abajo. Si no has cometido ningún fallo al introducir el programa verás aparecer el fondo, el castillo y los árboles. Si hay algún error repasa tus listados y compáralos con los del libro.



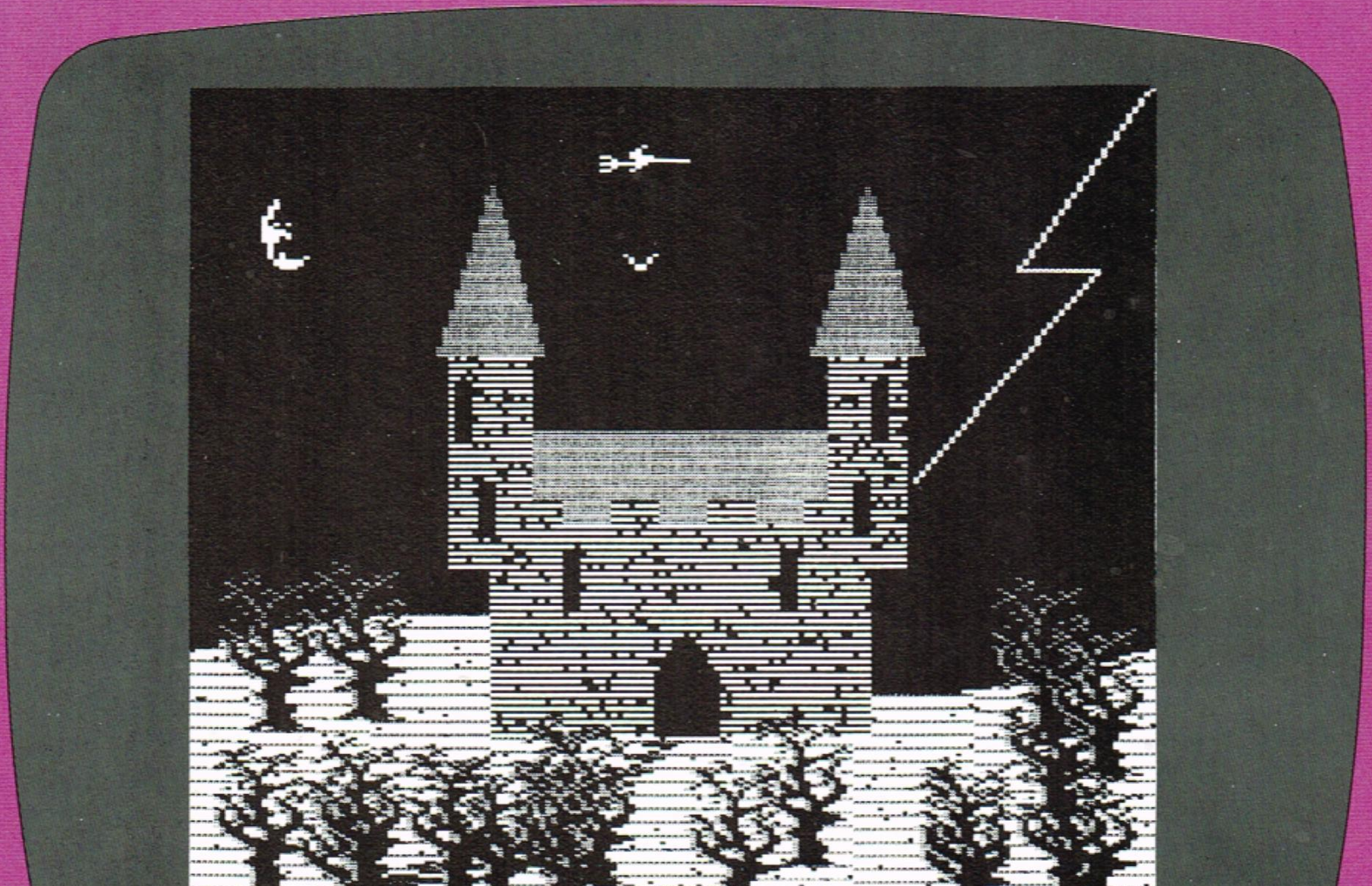
MSX : TECLEA **55**
GOTO 55 Y PRUEBA EL PROGRAMA.
PULSA **CTRL-STOP** UNA VEZ HAYAS COMPROBADO QUE EL PROGRAMA FUNCIONA CORRECTAMENTE.
BORRA LA LÍNEA 55 ANTES DE CONTINUAR CON LA SIGUIENTE SECCIÓN

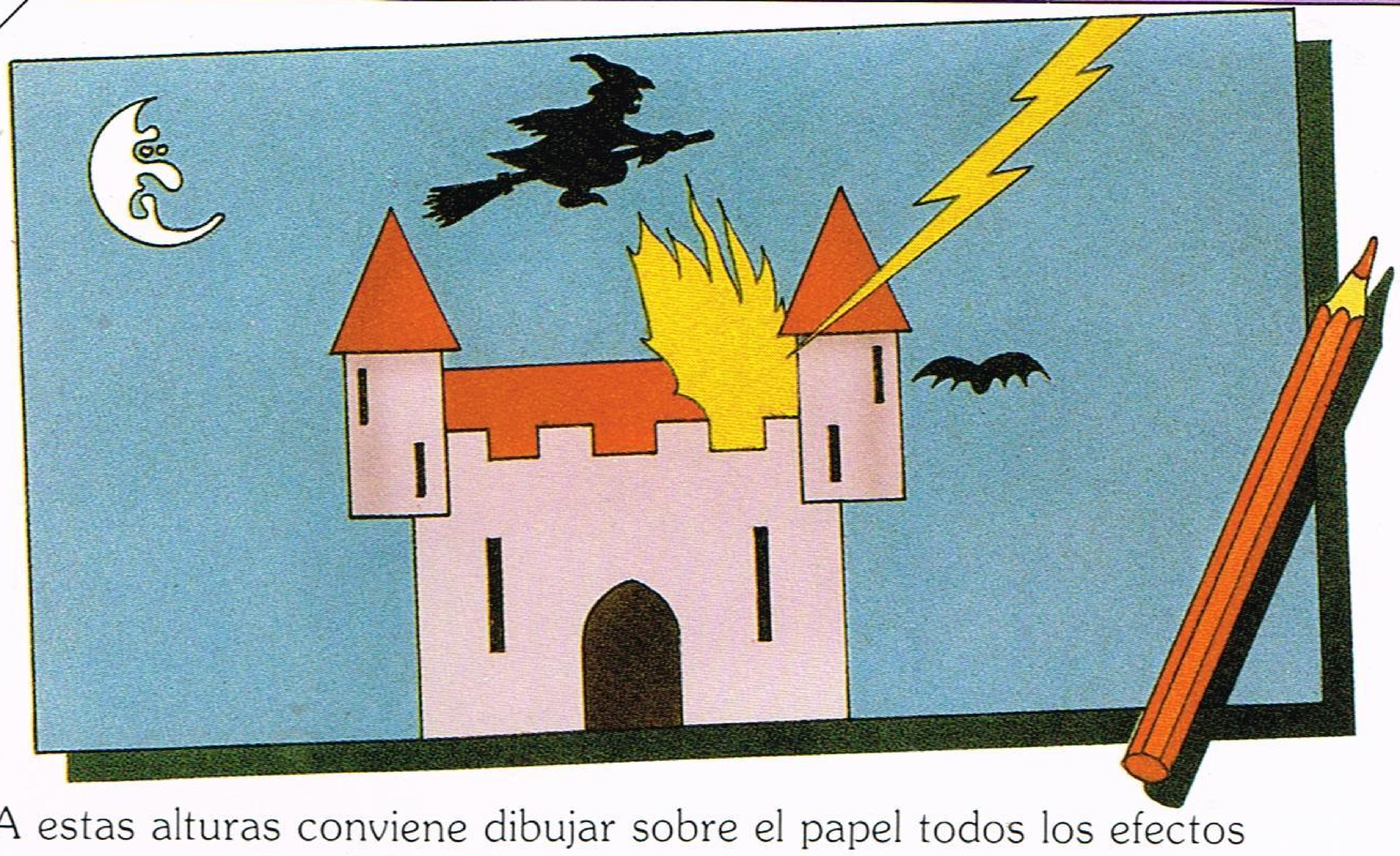


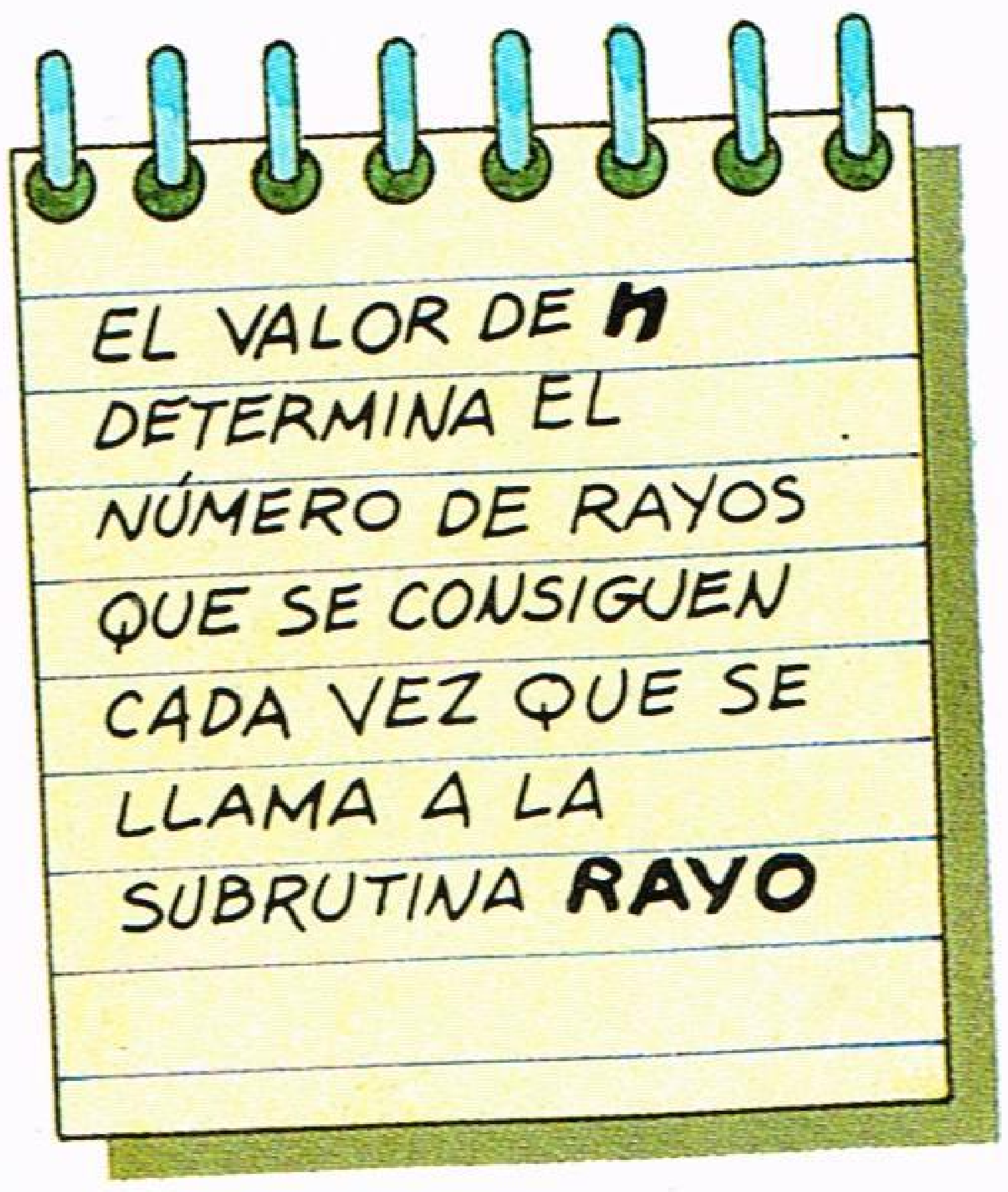
SPECTRUM : TECLEA **55**
GOTO 55, Y PRUEBA EL PROGRAMA. PULSA LAS TECLAS **CAPS SHIFT** Y **BREAK** UNA VEZ HAYAS COMPROBADO QUE EL PROGRAMA FUNCIONA CORRECTAMENTE.
BORRA LA LÍNEA 55 ANTES DE CONTINUAR CON LA SIGUIENTE SECCIÓN

MEDIANOCHE

Esta sección transforma el día en noche. El murciélago, la bruja, las llamas y la calavera son animados empleando técnicas diferentes según cada ordenador. Una vez que hayas aprendido estas técnicas puedes usarlas en tus programas.







EL VALOR DE **n** DETERMINA EL NÚMERO DE RAYOS QUE SE CONSIGUEN CADA VEZ QUE SE LLAMA A LA SUBRUTINA **RAYO**

A estas alturas conviene dibujar sobre el papel todos los efectos que vayas a introducir. Conoces las dimensiones de la casa, de forma que puedes distribuir todas las secuencias animadas a su alrededor. Plantéalo todo sobre el papel, así podrás estudiar si alguno de los caracteres colisiona con otro, o si se desplaza a algún lugar que no deba. Ten cuidado de no borrar los gráficos que ya has dibujado imprimiendo encima de ellos.

```
5000 REM Oscuridad
5010 PRINT USR(0)
5020 PUT SPRITE 4,(16,8),10
5030 R=1000:GOSUB 300
5040 RETURN
```

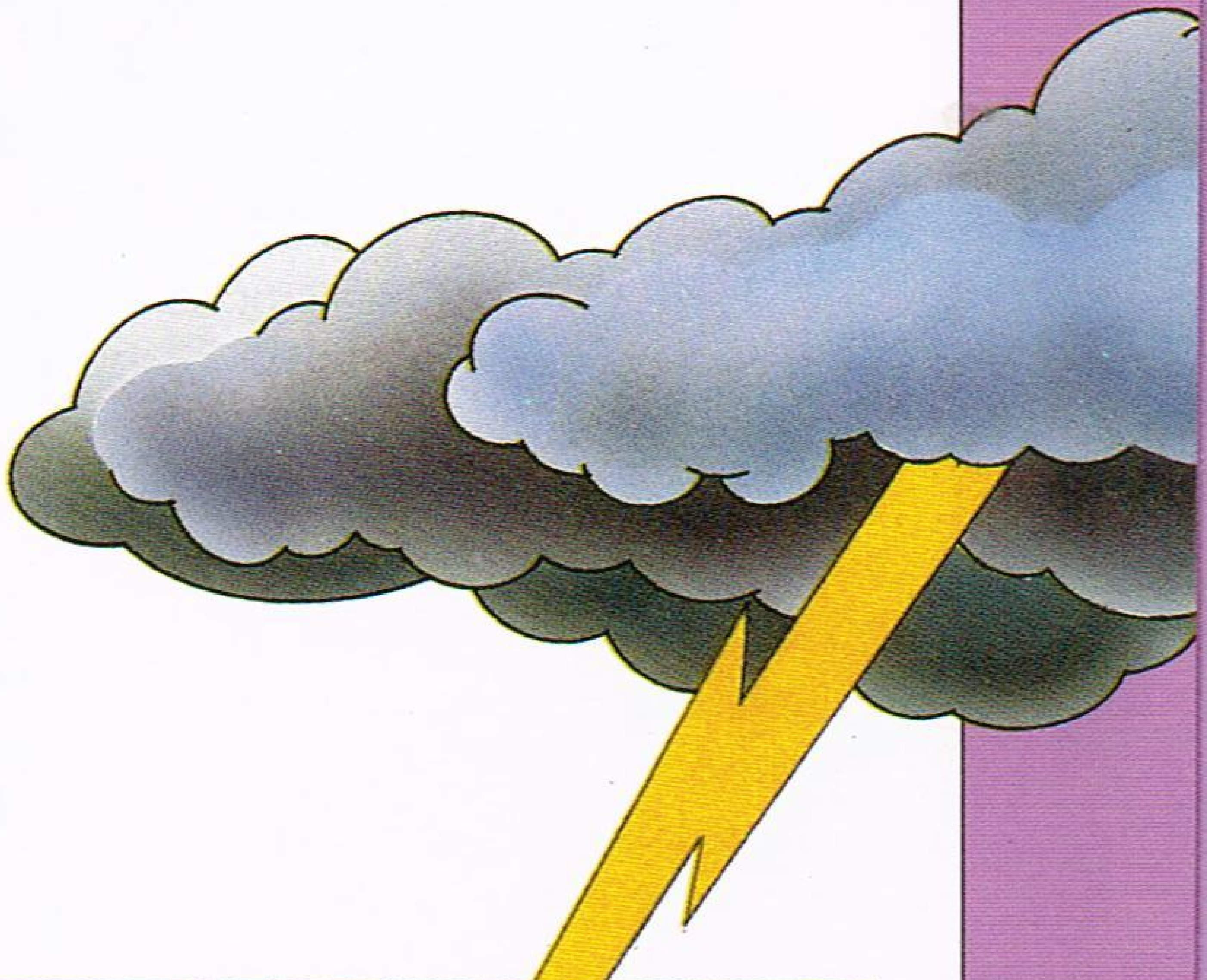
Para pasar de día a noche debes cambiar alguno de los colores de la pantalla. El **BASIC** del **MSX** no lo permite con un comando directo, de ahí que haya sido necesario introducir un programa en código máquina —mucho más rápido— que lo realice. El funcionamiento de este programa es muy simple; se limita a cambiar todas las posiciones de memoria de la **VRAM**

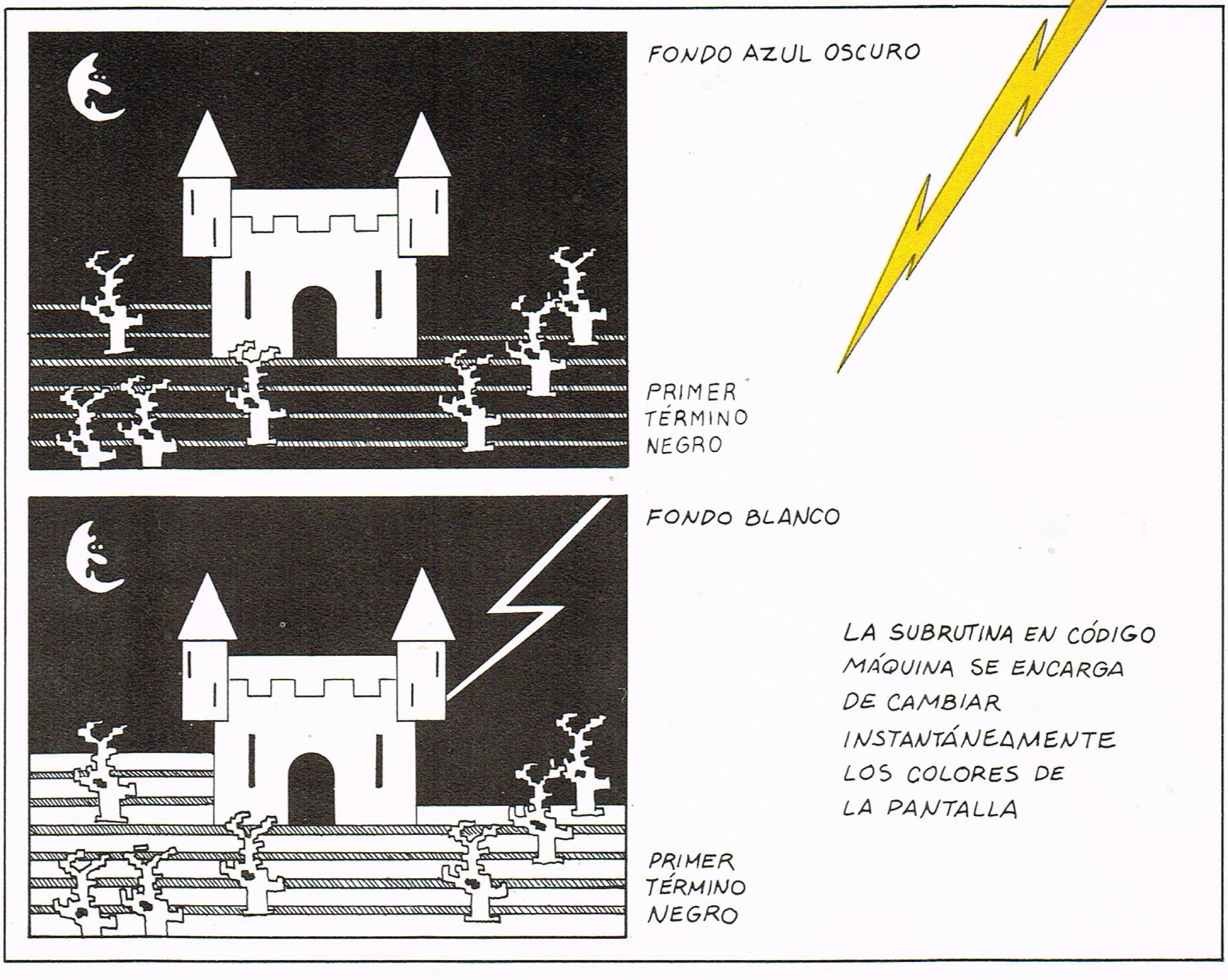
```
1500 REM Rayo
1510 FOR I=1 TO N
1520 PSET (240,0):DRAW "C11G32R16G31"
1530 POKE 61498!,&H1F:PRINT USR(0)
1540 R=200:GOSUB 300
1550 POKE 61498!,&H14:PRINT USR(0)
1560 PSET (240,0):DRAW "C1G32R16G31"
1570 R=400:GOSUB 300
1580 NEXT I
1590 RETURN
```

correspondientes a la tabla de colores e introducir en ellas el mismo valor, de forma que casi instantáneamente la rutina consigue que todos los colores de fondo pasen a azul oscuro y los correspondientes a primer término pasen a negro. No obstante puedes cambiar estos colores alterando la posición de memoria 61498 que contiene el valor a introducir en las posiciones de memoria anteriores, como luego veremos.

La línea 5010 pasa el control a esta rutina en código máquina, a continuación se imprime la luna en color amarillo colocando el **sprite** 4 en la posición oportuna. Por fin la línea 5030 introduce un nuevo retardo basado como antes en la subrutina 300.

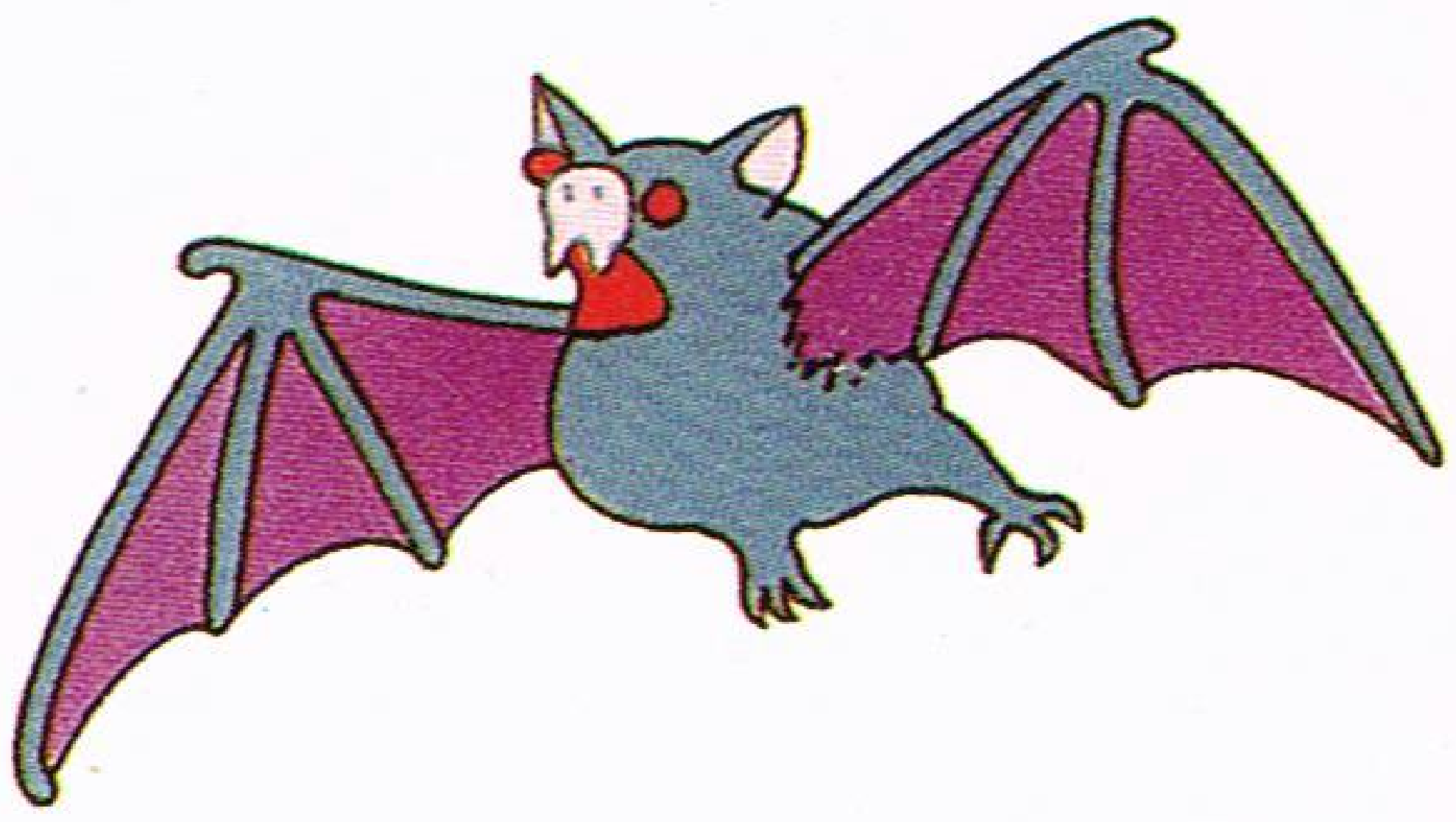
La subrutina **Rayo** produce un relámpago y dibuja un rayo cuatro veces seguidas —tantas como indique el parámetro **N**—. El rayo se consigue mediante la línea 1520, que dibuja una línea quebrada amarilla. A continuación la línea 1530 altera la posición de memoria que contiene los códigos de colores para la rutina en código máquina haciendo que el fondo pase a ser blanco —de código de color **F**— (15 en hexadecimal)— y el primer término negro —de código de color 1—; tras ésto se llama a la rutina en máquina. Después de un pequeño retardo la línea 1550 restablece los colores iniciales y la 1560 borra el rayo dibujándolo de nuevo en negro.





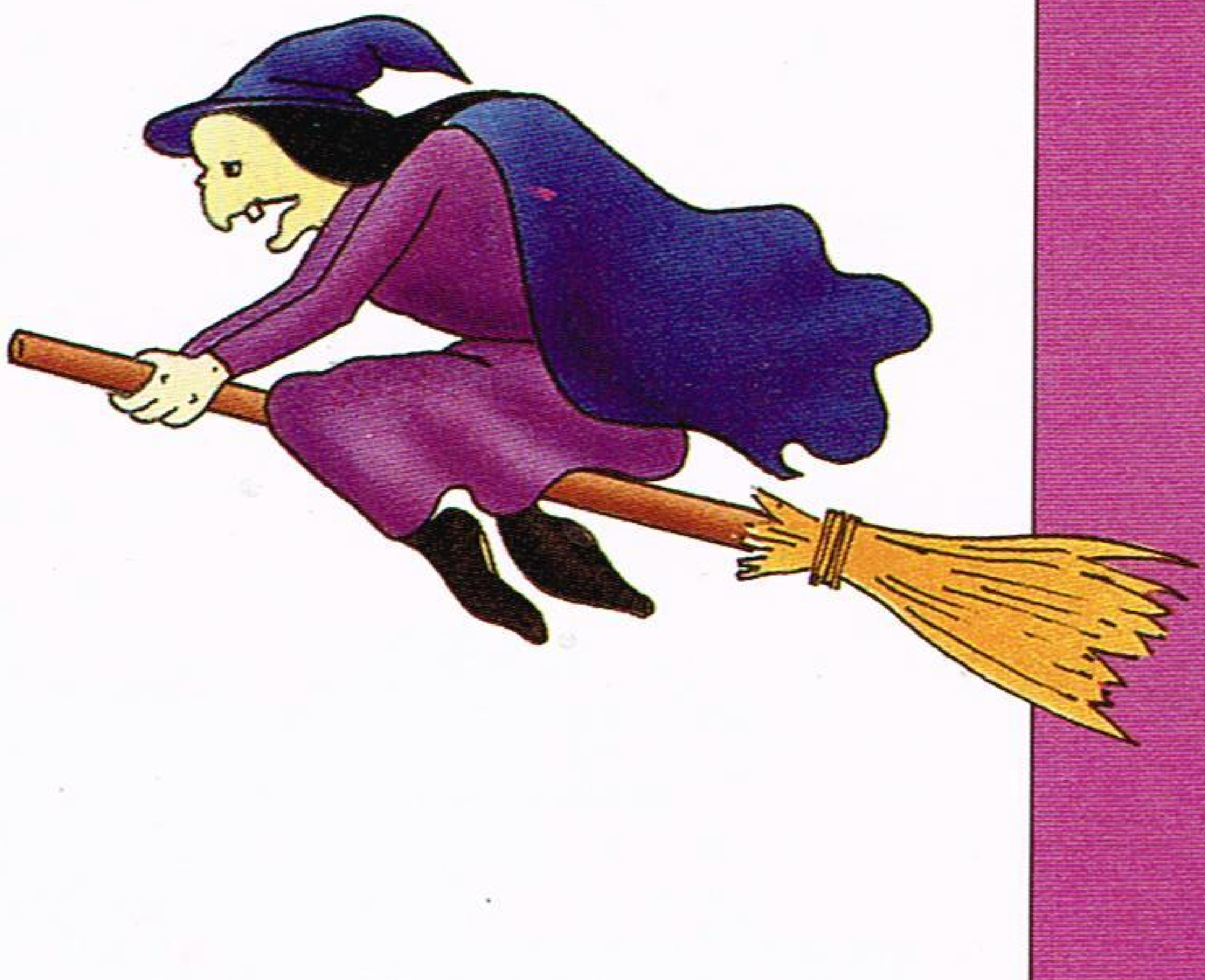
LA SUBRUTINA EN CÓDIGO MÁQUINA SE ENCARGA DE CAMBIAR INSTANTÁNEAMENTE LOS COLORES DE LA PANTALLA

```
500  REM Murcielago
510  IF M<50 THEN XM=XM+2:GOTO 530
520  M=M-15:A=M*3.14/180:XM=120+INT(40*SIN(A)):
     YM=30+INT (25*COS(A))
530  PUT SPRITE 1,(XM,YM),13
540  R=90:GOSUB 300:PUT SPRITE 1,(0,209)
550  PUT SPRITE 2,(XM,YM),13
560  R=20:GOSUB 300:PUT SPRITE 2,(0,209)
570  RETURN
```



El movimiento de la bruja y el murciélago lo controlan las líneas 80 a 130 del programa principal. La subrutina **Murciélago** —línea 500— emplea la variable **M** para calcular la trayectoria entre las dos torres. Inicialmente vale 1300 —línea 80— y cada vez que se ejecuta esta subrutina se decrementa 15 unidades. Cuando **M** supera cierto valor la línea 520 calcula sus coordenadas, que corresponden a las de una elipse, y actualiza el valor de **M**. Si no lo supera, la línea 510 habrá modificado **XM** para que el murciélago siga a la bruja. La línea 530 coloca el **sprite 1** —murciélago con las alas abiertas— en dicha posición y tras un retardo la línea 540 lo borra de la pantalla haciendo que su posición vertical valga 209. A continuación se repite el mismo proceso con el murciélago con las alas cerradas —líneas 550 y 560.

```
1000 REM Bruja
1010 YB=32+24*SIN(XB*2*3.14/254)
1020 PUT SPRITE 3,(XB,YB),3
1030 RETURN
```



El movimiento de la bruja es más simple pues su coordenada horizontal es incrementada de dos en dos en la línea 90 y su coordenada vertical se calcula en la línea 1010 de forma que trace una sinusoide en la pantalla. La línea 1020 coloca el **sprite** correspondiente en su posición. Cuando la bruja llega al final de la pantalla, la línea 120 del programa principal lo detecta y la hace desaparecer.

Para dibujar el fuego en la casa podríamos haber optado por definir el mismo **sprite** (llama) en diferentes planos y haber colocado estos **sprites** por toda la casa, teniendo en cuenta que no puede haber más de cuatro en la misma línea. No obstante se ha elegido un procedimiento más ingenioso, ya que con un sólo **sprite** podemos dar la sensación de que el fuego se extiende y la casa entera arde.

En primer lugar la línea 5510 **DIM**ensiona la matriz que va a contener las coordenadas x e y de cada llama (estas son tales que

```
5500 REM Fuego
5510 DIM F(2,6)
5520 FOR G=1 TO 6
5530 READ X,Y:F(1,G)=X:F(2,G)=Y
5540 NEXT G
5550 N=0
5560 FOR T=1 TO 6
5570 N=N+1
5580 FOR D=1 TO 30
5590 FOR J=1 TO N
5600 PUT SPRITE 5,(F(1,J),F(2,J)),11
5610 PUT SPRITE 5,(0,209)
5620 NEXT J:NEXT D:NEXT T
5630 FOR Y= 8 TO 96 STEP 8:FOR X=56 TO 200 STEP 8
5640 COLOR 1:PRESET (X,Y):PRINT#1,"["
5650 NEXT X:NEXT Y
```

el fuego parece salir de las ventanas). Las líneas 5520 a 5540 rellenan dicha matriz con los datos de la línea 7080. A continuación, tres bucles anidados se encargan de animar la escena. El fuego se imprime en amarillo en las posiciones dadas por **F(l,j)** y **F(2,j)** en la línea 5600, y la línea 5610 lo borra; ésto se repite 30 veces, de forma que las llamas titilan. Según aumenta **N**, **J** va tomando distintos valores, lo que hace aparecer una nueva llama en otra ventana, hasta que el fuego alcanza la última.

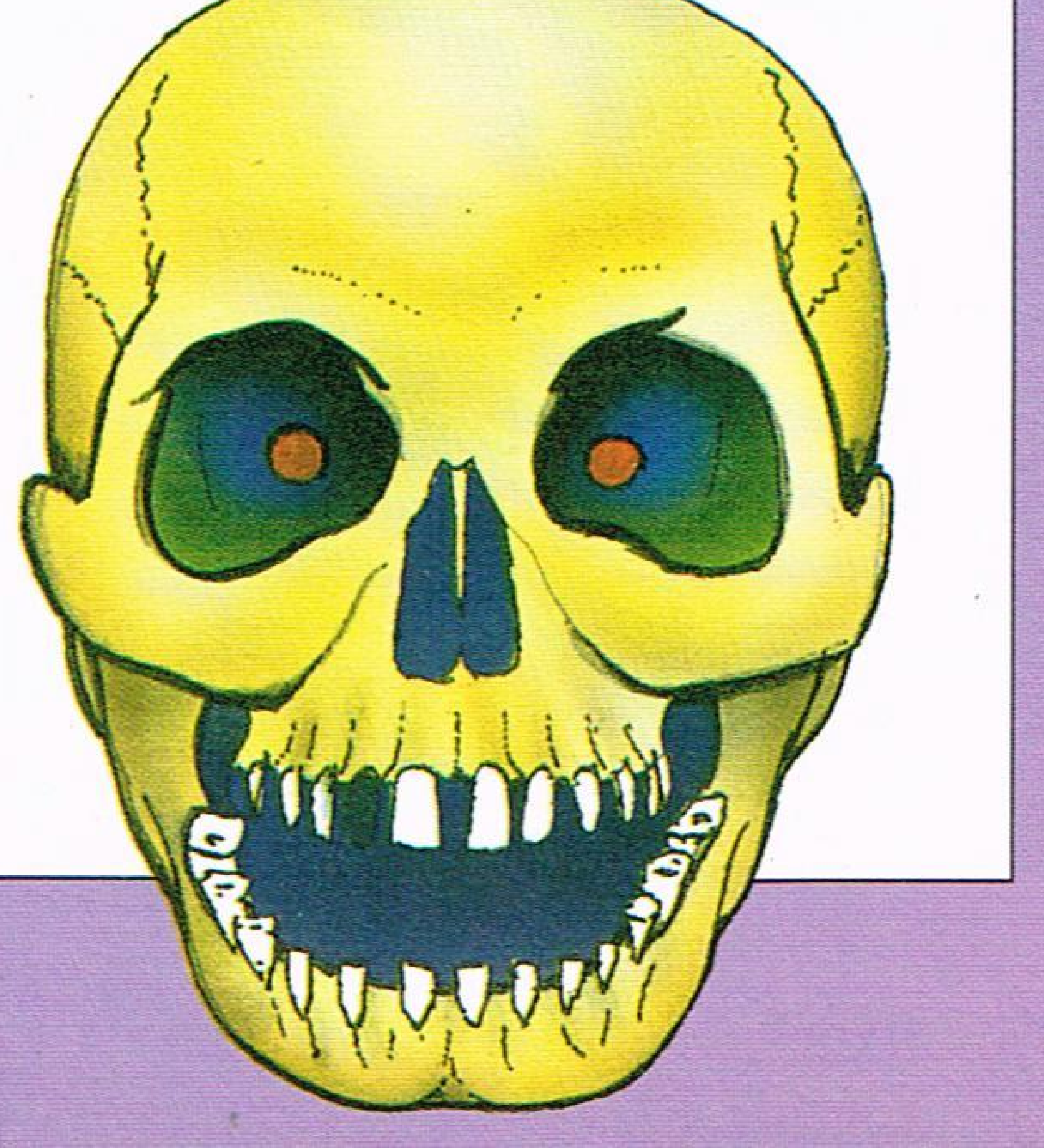
La casa se derrumba mediante el bucle 5630 a 5650, que imprime un rectángulo negro sobre la misma.

```
6000 REM Calavera
6010 COLOR 15
6020 CIRCLE (128,40),36,15,0,3.1459:PSET (92,38):
     DRAW"M+20,56R32M+20,-56"
6030 CIRCLE (112,40),8:CIRCLE(144,40),8
6040 PSET(110,64):DRAW "R4D8R4U4R4D4R4U4R4D4R4U4R4D4R4U
     8R4D12G4L4U4L4D4L4U4L4D4L4U4L4D4L4H4U12"
6050 PSET (126,44):DRAW "D16H4M+4,-12"
6060 PSET (130,44):DRAW "D16 E4M-4,-12"
6070 PAINT (128,5),15,15
6080 R=3000:GOSUB 300
6090 CLOSE #1
6100 RETURN
```

La fantasmagórica calavera que se cierne sobre las ruinas de la casa se consigue mediante la subrutina **Calavera.** Gracias al macrolenguaje gráfico que posee el sistema **MSX** podemos crear cualquier figura con facilidad. Así, la línea 6020 dibuja el contorno. Este se compone de un semicírculo y tres líneas rectas que unen sus extremos; todo se dibuja en color blanco (15). La línea 6030 dibuja los dos ojos y la 6040 la boca a base de una poligonal cerrada. Por último, las líneas 6050 y 6060 se encargan de trazar dos triángulos que simulan la nariz La sentencia **PAINT** (línea 6070) elige un punto interior a la figura y pinta el interior de la misma de blanco. Tras un retardo, la línea 6090 cierra el fichero que abrimos al principio del programa, mediante una sentencia **CLOSE.**

```
300  REM Retardo
310  FOR K=1 TO R:NEXT
320  RETURN
```

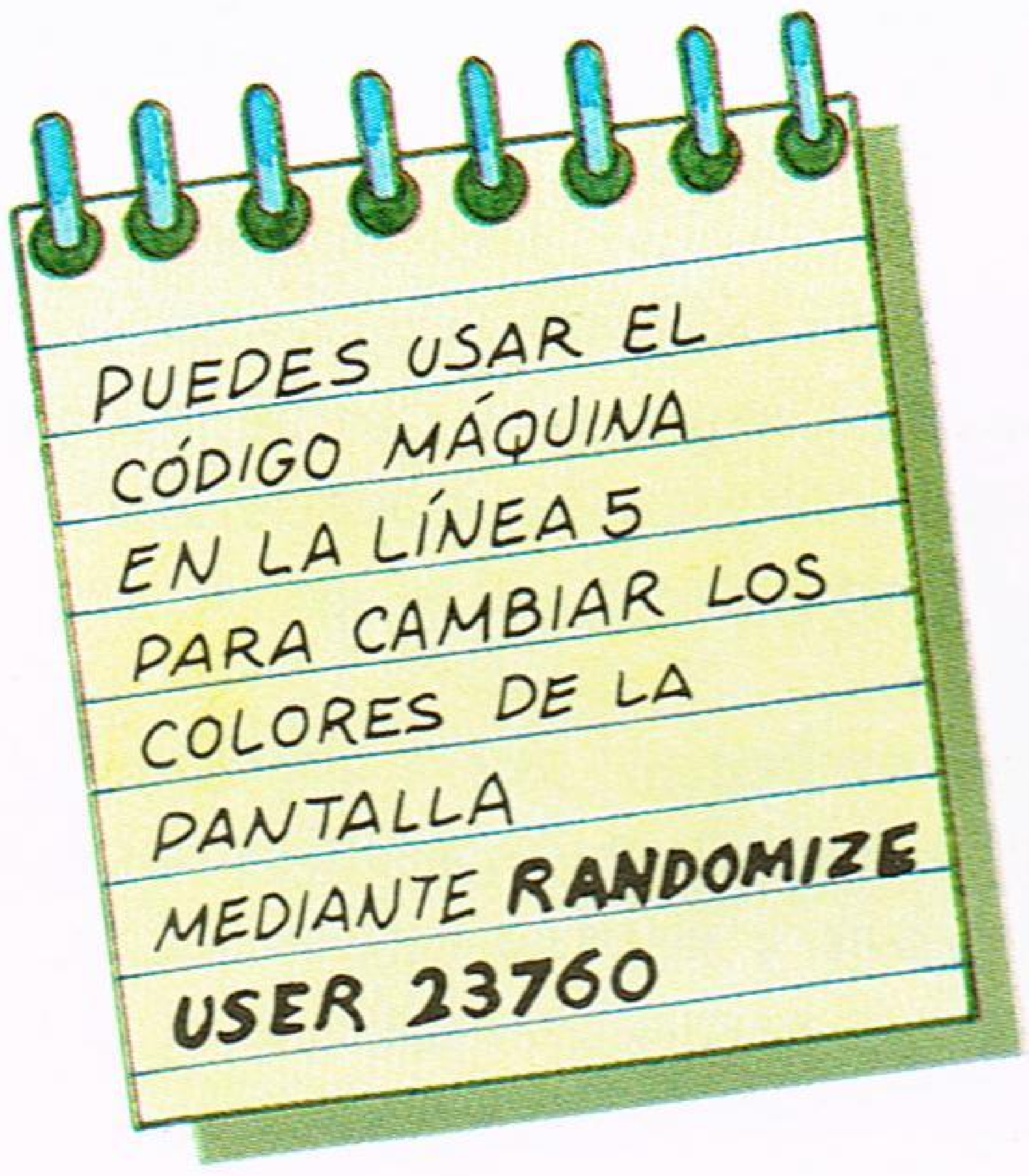
Unicamente queda analizar la subrutina **Retardo** que, como habíamos apuntado, se limita a contar de 1 a **R,** con lo que el tiempo que tardará en ejecutarse será proporcional al valor de esta variable.



```
5000 REM oscuridad
5010 POKE 23761,1: RANDOMIZE USR 23760
5020 PAPER 0: INK 6: PRINT AT 3,3;"G";AT 4,3;"H":
     REM (G);(H)
5030 PAUSE 100
5040 RETURN
```

Para pasar de día a noche hay que cambiar algunos colores de la pantalla. El **BASIC** del Spectrum no lo permite, de ahí que haya que emplear el código máquina que habíamos introducido. El programa en código máquina se activa en la línea 5010; todos los pixels correspondientes a tinta se cambian a azul oscuro, y los correspondientes a papel a negro. **RANDOMIZE USR** significa "ejecuta el programa en código máquina" que comienza en la dirección de memoria indicada. La línea 5020 imprime la luna en color amarillo y, tras una pequeña pausa, la subrutina finaliza.

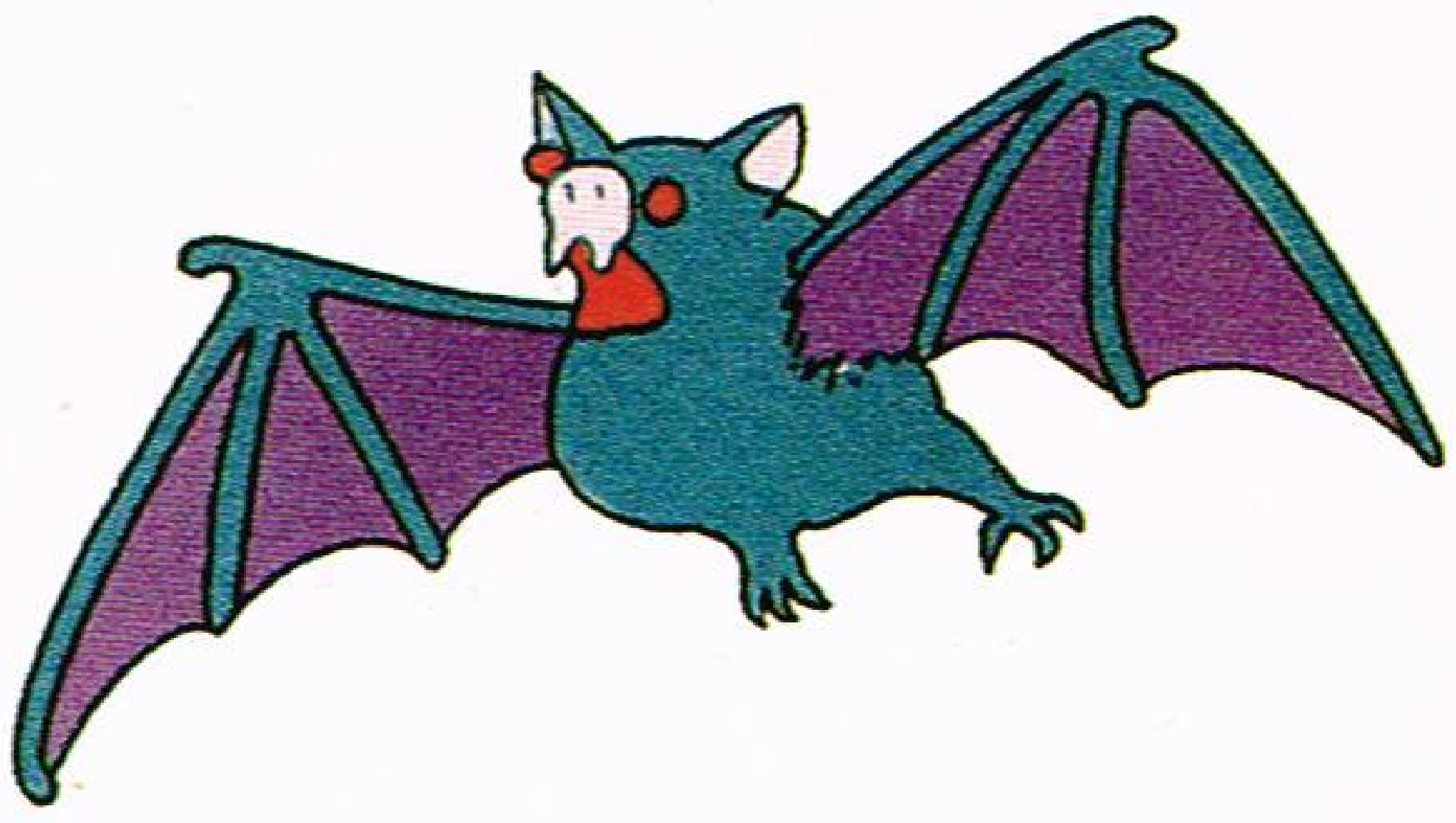
El programa de control llama ahora a la subrutina **rayo** que comienza en la línea 1500. Una serie de instrucciones **PLOT** y **DRAW** dibujan una línea negra quebrada desde la esquina superior derecha de la pantalla. El rayo no se ve hasta que la línea 1530 llama al programa en código máquina, que cambia los pixels del rayo a blanco **(POKE 23761,7).** El código máquina cambia el color de la luna, de ahí que la línea 1540 la imprima de nuevo en amarillo. Esto ocurre tan rápidamente que la luna parece no cambiar de color. Después de una pequeña pausa en la línea 1545 la rutina en máquina vuelve a cambiar el color de la tinta a azul. La línea 1560 restablece la luna en amarillo y el rayo se borra imprimiéndolo de nuevo en negro. Tras una pequeña pausa se repite el bucle otras cuatro veces, ya que habíamos fijado la variable **n** a cinco.



PUEDES USAR EL CÓDIGO MÁQUINA EN LA LÍNEA 5 PARA CAMBIAR LOS COLORES DE LA PANTALLA MEDIANTE **RANDOMIZE USER 23760**

```
1500 REM rayo
1510 FOR 1=1 TO n
1520 INK 0: PLOT 255,170: DRAW -39,-40: DRAW 10,0:
     DRAW -29,-40
1530 POKE 23761,7: RANDOMIZE USR 23760
1540 INK 6: PRINT AT 3,3;"G";AT 4,3 ;"H": REM (G);(H)
1545 PAUSE 5
1550 POKE 23761,1: RANDOMIZE USR 23760
1560 INK 6: PRINT AT 3,3;"G";AT 4,3;"H": REM (G);(H)
1570 INK 0: PLOT 255,170: DRAW -39,-40: DRAW 10,0:
     DRAW -29,-40
1575 PAUSE 1
1580 NEXT 1
1590 RETURN
```

```
500 REM murcielago
510 IF m<150 THEN  LET xm=xm+1: GO TO 530
520 LET m=m-8: LET a=m*PI/180:
    LET xm=16+INT (4.5*SIN (a))
    LET ym=4+INT (3.5*COS (a))
530 PAPER 0: INK 7: PRINT AT ym,xm;"R": REM (R)
535 PAUSE 3
540 PRINT AT ym,xm;"S": REM (S)
545 PAUSE 3
550 PRINT AT ym,xm;" ": REM esp
560 RETURN
```



Si analizas las líneas 90 a 130 del programa de control (que corresponden a un bucle **FOR... NEXT)**, verás que el murciélago y la bruja son animados simultáneamente. La subrutina **murciélago**, que comienza en la línea 500, emplea la variable **m** para calcular las coordenadas de la trayectoria elíptica que sigue el murciélago entre las dos torres.

Sin embargo, antes de realizar ningún cálculo, la línea 510 estudia el valor de **m**. Si éste es menor que 150 el murciélago deja de dar vueltas y parece intentar cazar a la bruja. Si **m** no es menor que 150, la subrutina salta a la línea 520, donde se emplean las funciones SENO y COSENO para dibujar el camino espiral. A continuación se imprime el murciélago batiendo sus alas alternando los caracteres **R** y **S**.

La bruja se anima de distinta forma al murciélago mediante la subrutina que comienza en la línea 1000. Esta subrutina opera de la misma forma que la rutina que animaba el avión al comienzo del libro, mediante diferentes caracteres definidos por el usuario impresos consecutivamente en cada posición de carácter a través de la pantalla.

Antes de imprimir cada grupo de caracteres correspondientes a la bruja se llama a la subrutina **murciélago.** De esta forma conseguimos que el movimiento de aquella sea mucho más suave, ya que mientras el murciélago se mueve un carácter la bruja sólo lo hace dos pixels. De no ser así el movimiento de la bruja sería mucho más brusco.

```
1000 REM bruja
1010 PAPER 0: INK 7
1020 PRINT AT 1,b;" IJ": GO SUB 500: REM esp,(I),(J)
1030 PRINT AT 1,b;" KL": GO SUB 500: REM esp,(K),(L)
1040 PRINT AT 1,b;" MN": GO SUB 500: REM esp,(M),(N)
1050 PRINT AT 1,b;" OPQ": REM esp,(O),(P),(Q)
1060 RETURN
```

```
5500 REM fuego
5510 DIM f(2,6): INK 6: PAPER 0
5520 FOR g=1 TO 6
5530 READ y,x: LET f(1,g)=y: LET f(2,g)=x
5540 NEXT g
5550 LET n=0
5560 FOR t=1 TO 6
5570 LET n=n+1
5580 FOR d=1 TO 30
5590 FOR j=1 TO n
5600 PRINT AT f(1,j),f(2,j);"T": REM (t)
5610 PRINT AT f(1,j),f(2,j);" ": REM espacio
5620 NEXT j: NEXT d: NEXT t
5630 FOR y=2 TO 11: FOR x=7 TO 23
5640 PRINT AT y,x;" ": REM espacio
5650 NEXT x: NEXT y
5660 RETURN
```

La animación de las llamas es bastante ingeniosa. Uno de los problemas es que el ordenador no puede tener más de un color de tinta y otro de papel en cada carácter. Para resolverlo haremos aparecer el fuego desde dentro del castillo, únicamente en las ventanas.

La línea 5510 **DIM**ensiona una matriz, que guardará las posiciones de cada carácter correspondiente a una llama (T); estas posiciones se ajustan a las coordenadas de las ventanas. Las líneas 5560 a 5620 son tres bucles **FOR... NEXT** anidados. En primer lugar se imprime la llama en amarillo en la posición **f(1,1)** cuando **j** es igual a uno en la línea 5600. La línea 5610 imprime un espacio sobre el fuego. Este proceso se repite 30 veces haciendo vacilar la llama. Según aumenta **n, j** toma un nuevo valor, y el proceso se repite en cada ventana hasta que las llamas alcanzan la última.

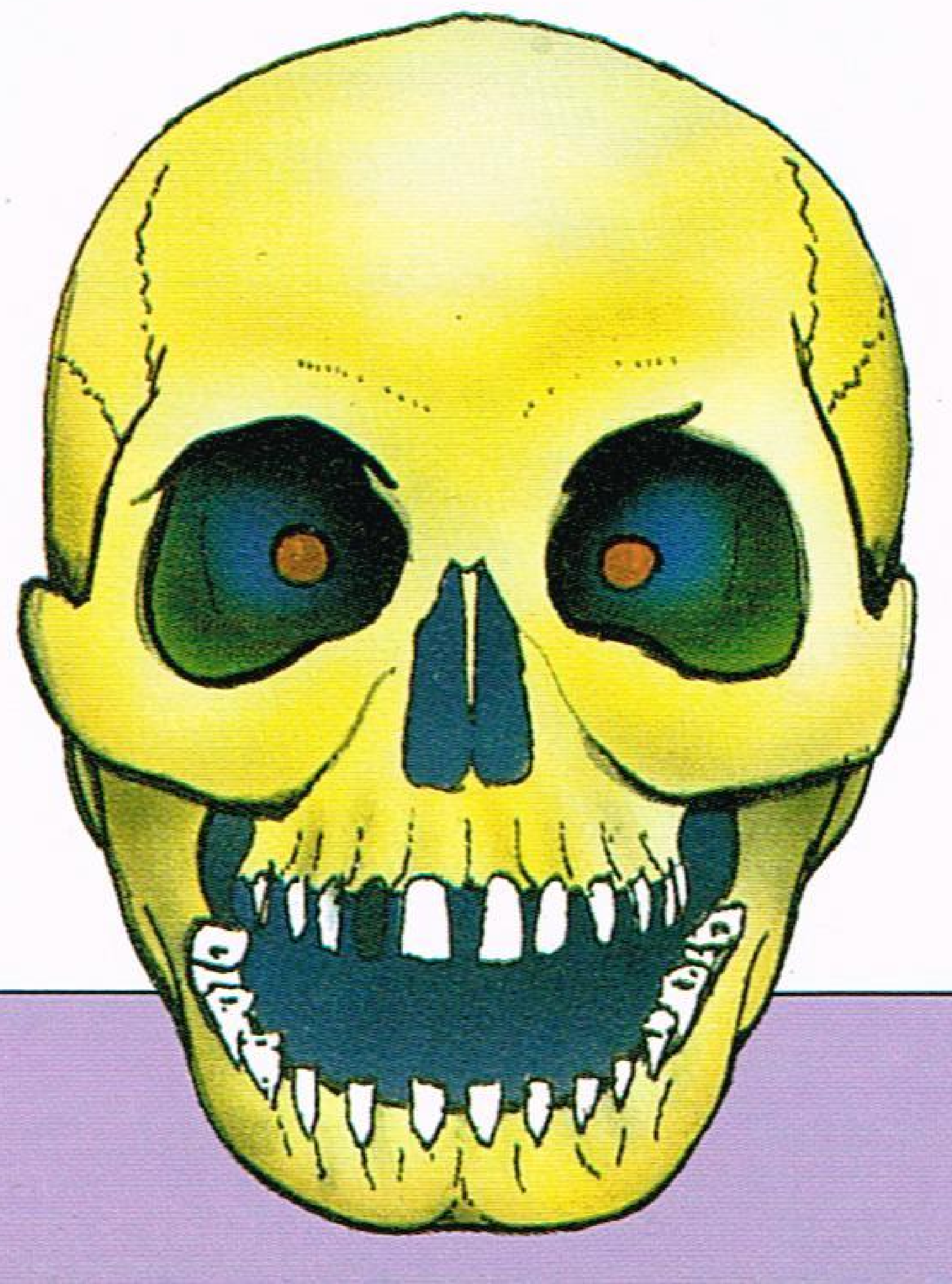
La casa se derrumba mediante dos bucles **FOR... NEXT** anidados en las líneas 5630 a 5650 que borran la casa con una serie de espacios en negro.

```
6000 REM calavera
6010 INK 0
6020 FOR s=270 TO 450 STEP 2
6030 LET a=s*PI/180: LET x=40*SIN (a)
6040 LET y=40*COS (a): PLOT 127,135: DRAW x,y
6050 NEXT s: LET x1=87: LET x2=80
6060 FOR y=135 TO 87 STEP -1
6070 PLOT x1,y: DRAW x2,0: LET x1=x1+.3: LET x2=x2-.6
6080 NEXT y
6090 PAPER 0: PRINT AT 6,13;"  ";AT 6,17;"  ":
     REM esp,esp;esp,esp
6100 PRINT AT 8,15;"  ": REM espacio,espacio
6110 PRINT AT 10,14;"   ": REM espacio,espacio,espacio
6120 FOR y=81 TO 169 STEP 8: FOR x=87 TO 167 STEP 8
6130 IF POINT (x,y)=1 THEN  PAPER 0:
     INK INT (RND*(7))+1: FLASH 1:
     PRINT AT 22-y/8,x/8-1;"6": FLASH 0: REM (6)
6140 NEXT x: NEXT y
6150 RETURN
```

La fantasmagórica calavera que se cierne sobre la casa quemada se crea mediante la subrutina **calavera.** La parte superior es un semicírculo que se dibuja en color negro **(INK 0)** mediante el lazo que comienza en la línea 6020, de forma que no se ve en esta etapa. El resto de la cara se consigue mediante otro lazo en las líneas 6060 a 6080 que dibuja una serie de líneas horizontales que se van haciendo cada vez más estrechas según nos acercamos a las ruinas de la casa.

La línea 6090 coloca los ojos con dos espacios en negro, y las líneas 6100 y 6110 hacen lo mismo con la nariz y la boca.

Recuerda que hasta ahora la cara es invisible, ya que está coloreada de negro sobre el cielo del mismo color. Para conseguir un efecto intermitente emplearemos la función **POINT** que estudia ciertos pixels sobre el área de la calavera. Si el pixel corresponde a tinta —en otras palabras, si **POINT** devuelve el valor 1— entonces se imprime el carácter gráfico 6 en esa posición con un color aleatorio y de forma intermitente **(FLASH 1).** La calavera parece emerger de las ruinas de la casa según el lazo la estudia desde la parte inferior hasta la más alta.



Almacena tus programas

A continuación se da el listado completo del programa tanto para MSX como para Spectrum. Compruébalos si aún tienes problemas. El listado MSX contiene líneas que no aparecen en el libro (las que terminan en 5). Estas se han añadido para conseguir efectos especiales en la ejecución del programa. Hacen que éste sea más tétrico y no lo retardan.

Listados de los programas

MSX

```
   1 REM Casa Encantada
   5 CLEAR 200,61482!
  10 SCREEN 2,2:COLOR 1,3,1:CLS
  15 ON SPRITE GOSUB 8005:SPRITE ON
  20 GOSUB 6500:REM Inicializar
  30 GOSUB 2000:REM Fondo
  40 GOSUB 2500:REM Castillo
  50 GOSUB 4500:REM Arboles
  60 GOSUB 5000:REM Oscuridad
  70 N=4:GOSUB 1500:REM Rayo
  80 M=1300
  85 BEEP
  90 FOR XB=0 TO 254 STEP 2
 100 GOSUB 500:REM Murcielago
 110 GOSUB 1000:REM Bruja
 120 IF XB=254 THEN PUT SPRITE 3,(0,209)
 130 NEXT XB
 140 N=4:GOSUB 1500:REM Rayo
 150 GOSUB 5500:REM Fuego
 160 GOSUB 6000:REM Calavera
 170 END
 300 REM Retardo
 310 FOR K=1 TO R:NEXT
 320 RETURN
 500 REM Murcielago
 510 IF M<50 THEN XM=XM+2:GOTO 530
 520 M=M-15:A=M*3.14/180:XM=120+INT(40*SIN(A)):
     YM=30+INT (25*COS(A))
 530 PUT SPRITE 1,(XM,YM),13
 540 R=90:GOSUB 300:PUT SPRITE 1,(0,209)
 545 PLAY "T155L64S8M500O1G"
 550 PUT SPRITE 2,(XM,YM),13
 560 R=20:GOSUB 300:PUT SPRITE 2,(0,209)
 565 PLAY "G"
 570 RETURN
1000 REM Bruja
1010 YB=32+24*SIN(XB*2*3.14/254)
1020 PUT SPRITE 3,(XB,YB),3
1030 RETURN
1500 REM Rayo
1510 FOR I=1 TO N
1520 PSET (240,0):DRAW "C11G32R16G31"
1530 POKE 61498!,&H1F:PRINT USR(0)
1535 SOUND 0,0:SOUND 1,0:SOUND 2,0:SOUND 3,0:SOUND 4,0:
     SOUND 5,0:SOUND 6,31:SOUND 7,7:SOUND 8,16:SOUND 9,
     16:SOUND 10,16:SOUND 11,0:SOUND 12,60:SOUND 13,0
1540 R=200:GOSUB 300
1550 POKE 61498!,&H14:PRINT USR(0)
1560 PSET (240,0):DRAW "C1G32R16G31"
1570 R=400:GOSUB 300
1580 NEXT I
1590 RETURN
2000 REM Fondo
2010 FOR R=1 TO 100
```

Continuación MSX

```
2020 X=INT (RND(1)*255):Y=130+INT (RND(1)*60)
2030 PSET (X,Y),1:NEXT
2040 OPEN "GRP:" AS #1:COLOR 5
2050 FOR Y=0 TO 120 STEP 8:FOR X=0 TO 248 STEP 8
2060 PRESET (X,Y):PRINT#1,"■"
2070 NEXT X:NEXT Y
2080 I=16
2090 FOR P=63 TO 0 STEP -2
2100 X=138*SIN(P/60):Y=192-P:LINE (X,Y)-(X+I,Y),1
2110 I=I+1.5:NEXT
2120 RETURN
2500 REM Castillo
2510 X1=80:Y1=72:X2=168:Y2=120:LAD=300:
     GOSUB 3000:REM Bloque frontal
2520 X1=64:Y1=40:X2=80:Y2=80:LAD=70:
     GOSUB 3000:REM Torre izquierda
2530 X1=168:Y1=40:X2=184:Y2=80:LAD=70:
     GOSUB 3000:REM Torre derecha
2540 XS=76:YS=8:GOSUB 3500:REM Tejado izquierdo
2550 XS=180:YS=8:GOSUB 3500:REM Tejado derecho
2560 FOR Y=60 TO 72
2570 LINE (89,Y)-(167,Y),6
2580 NEXT
2600 X=68:Y=47:GOSUB 4000
2610 X=76:Y=63:GOSUB 4000
2620 X=182:Y=47:GOSUB 4000
2630 X=174:Y=63:GOSUB 4000
2640 X=96:Y=83:GOSUB 4000
2650 X=154:Y=83:GOSUB 4000
2660 FOR Y=104 TO 127:IF Y<112 THEN
     LINE (232-Y,Y)-(24+Y,Y),1:NEXT
     ELSE LINE (120,Y)-(136,Y),1:NEXT
2670 RETURN
3000 REM Bloque
3010 FOR Y=Y1 TO Y2 STEP 8:FOR X=X1 TO X2 STEP 8
3020 COLOR 9:PRESET (X,Y):PRINT #1,"■"
3030 NEXT X:NEXT Y
3040 FOR Y=Y1 TO Y2+7 STEP 2
3050 LINE (X1,Y)-(X2+7,Y),1
3060 NEXT Y
3070 COLOR 1:FOR M=1 TO LAD
3080 A=INT (RND(1)*(X2-X1+7)+X1):
     B=INT (RND(1)*(Y2-Y1+7)+Y1):PSET (A,B),1
3090 NEXT M
3100 RETURN
3500 REM Segmento
3510 R=38
3520 FOR T=160 TO 200 STEP 2
3530 A=T*3.14/180:X=R*SIN(A):Y=R*COS(A)
3540 LINE (XS,YS)-(XS+X,YS-Y),6
3550 NEXT
3560 RETURN
4000 REM Ventana
4010 I=5
4020 FOR J=Y TO Y+17:IF J<Y+5 THEN
     LINE (X+I/2,J)-(X+6-I/2,J),1:I=I-1:NEXT
     ELSE LINE (X,J)-(X+6,J),1:NEXT
4030 RETURN
4500 REM Arboles
4510 DIM A(21)
4520 FOR J=1 TO 21:READ N:A(J)=N:NEXT
4530 FOR J=1 TO 21
4540 I=BASE(10)+A(J)
4550 VPOKE (I),0:VPOKE (I+1),1:VPOKE (I+32),32:
     VPOKE (I+33),33:VPOKE (I+64),64:VPOKE (I+65),65
4560 NEXT J
4570 R=1000:GOSUB 300
4580 RETURN
5000 REM Oscuridad
5010 PRINT USR(0)
5020 PUT SPRITE 4,(16,8),10
5030 R=1000:GOSUB 300
5040 RETURN
5500 REM Fuego
5510 DIM F(2,6)
```

Continuación MSX

```
5520 FOR G=1 TO 6
5530 READ X,Y:F(1,G)=X:F(2,G)=Y
5540 NEXT G
5550 N=0
5560 FOR T=1 TO 6
5570 N=N+1
5580 FOR D=1 TO 30
5590 FOR J=1 TO N
5600 PUT SPRITE 5,(F(1,J),F(2,J)),11
5605 FOR W=1 TO INT(RND(-TIME)*30):NEXT W:
     Z=INT(RND(-TIME)*150)+2:SOUND 0,Z:SOUND 1,15:
     SOUND 6,5:SOUND 7,&B10110110:SOUND 8,16:SOUND 12,2
     :SOUND 13,4
5610 PUT SPRITE 5,(0,209)
5620 NEXT J:NEXT D:NEXT T
5630 FOR Y= 8 TO 96 STEP 8:FOR X=56 TO 200 STEP 8
5640 COLOR 1:PRESET (X,Y):PRINT#1,"["
5650 NEXT X:NEXT Y
6000 REM Calavera
6010 COLOR 15
6020 CIRCLE (128,40),36,15,0,3.1459:PSET (92,38):
     DRAW"M+20,56R32M+20,-56"
6030 CIRCLE (112,40),8:CIRCLE(144,40),8
6040 PSET(110,64):DRAW "R4D8R4U4R4D4R4U4R4D4R4U4R4D4R4U
     8R4D12G4L4U4L4D4L4U4L4D4L4U4L4D4L4H4U12"
6050 PSET (126,44):DRAW "D16H4M+4,-12"
6060 PSET (130,44):DRAW "D16 E4M-4,-12"
6070 PAINT (128,5),15,15
6075 BEEP:PLAY "T90S1M5000O1A4A8.A32A4O2C8.O1B16B8.A16A
     8.A16A2"
6080 R=3000:GOSUB 300
6090 CLOSE #1
6100 RETURN
6500 REM Inicializar
6510 FOR I=61483! TO 61508!
6520 READ N$:POKE I,VAL("&H"+N$):NEXT
6530 DEFUSR=61483!
6540 FOR I=BASE(12) TO BASE(12)+15:READ N:
     VPOKE (I+4096),N:NEXT
6550 FOR I=BASE(12) TO BASE(12)+15:READ N:
     VPOKE (I+4352),N:NEXT
6560 FOR I=BASE(12) TO BASE(12)+15:READ N:
     VPOKE (I+4608),N:NEXT
6570 FOR I=BASE (14)+32 TO BASE(14)+191:READ N:
     VPOKE I,N:NEXT
6580 RETURN
7000 DATA DB,99,21,00,20,0E,99,ED,69,ED,61,21,00,00,3E,
     14,D3,98,23,7C,FE,18,C2,39,F0,C9:REM Cod. maquina
7010 DATA 16,129,73,40,16,80,48,9,128,32,74,134,164,68,
     204,30,5,38,178,27,31,15,3,3,48,96,192,192,194,244
     ,152,128,1,1,1,3,3,7,31,51,192,192,192,128,128,128
     ,224,252:REM Arbol
7020 DATA 0,0,0,16,56,125,255,63,15,3,2,0,0,0,0,0,0,0,0
     ,8,28,190,255,252,240,192,64,0,0,0,0,0:REM Murc.1
7030 DATA 0,0,0,0,0,13,31,63,63,63,26,24,8,0,0,0,0,0,0,
     0,0,176,248,252,252,252,88,24,16,0,0,0:REM Murc.2
7040 DATA 7,1,0,0,0,1,3,159,207,255,195,129,1,0,0,0,128
     ,192,96,96,224,192,192,224,240,255,192,128,128,192
     ,0,0:REM Bruja
7050 DATA 3,14,28,56,122,120,254,255,255,248,248,96,102
     ,55,31,3,0,0,0,0,128,0,0,128,128,0,0,4,12,152,248,
     240:REM Luna
7060 DATA 0,32,32,34,50,51,51,127,125,236,228,100,53,31
     ,7,7,0,0,17,17,49,115,227,167,167,191,186,164,200,
     208,224,224:REM Llama
7070 DATA 521,535,540,547,550,562,565,570,580,592,600,
     610,637,640,657,661,667,686,691,696,700:
     REM Posicion arboles
7080 DATA 177,48,169,64,149,84,91,84,71,64,63,48:
     REM Posicion llamas
8005 FOR R=85 TO 250 STEP 15;FOR T=1 TO 100 STEP 20:
     FOR A=1 TO T+15 STEP -15:SOUND 0,R-T:SOUND 1,0:
     SOUND 7,254:SOUND 8,16:SOUND 12,5:SOUND 13,1:
     NEXT A:NEXT T:NEXT R:SPRITE OFF:BEEP:RETURN
```

SPECTRUM

```
   5 REM ******************
  10 REM casa encantada
  11 REM 18 bytes de codigo maquina almacenados en la
        linea 5
  20 GO SUB 6500: REM inicializar
  30 GO SUB 2000: REM fondo
  40 GO SUB 2500: REM castillo
  50 GO SUB 4500: REM arboles
  60 GO SUB 5000: REM oscuridad
  70 LET n=5: GO SUB 1500: REM rayo
  80 LET m=1300
  90 FOR b=0 TO 29
 100 GO SUB 500: REM murcielago
 110 GO SUB 1000: REM bruja
 120 IF b=29 THEN  PRINT AT 1,29;"   ": REM esp,esp,esp
 130 NEXT b
 140 LET n=5: GO SUB 1500: REM rayo
 150 GO SUB 5500: REM fuego
 160 GO SUB 6000: REM calavera
 170 STOP
 500 REM murcielago
 510 IF m<150 THEN  LET xm=xm+1: GO TO 530
 520 LET m=m-8: LET a=m*PI/180:
     LET xm=16+INT (4.5*SIN (a))
     LET ym=4+INT (3.5*COS (a))
 530 PAPER 0: INK 7: PRINT AT ym,xm;"R": REM (R)
 535 PAUSE 3
 540 PRINT AT ym,xm;"S": REM (S)
 545 PAUSE 3
 550 PRINT AT ym,xm;" ": REM esp
 560 RETURN
1000 REM bruja
1010 PAPER 0: INK 7
1020 PRINT AT 1,b;" IJ": GO SUB 500: REM esp,(I),(J)
1030 PRINT AT 1,b;" KL": GO SUB 500: REM esp,(K),(L)
1040 PRINT AT 1,b;" MN": GO SUB 500: REM esp,(M),(N)
1050 PRINT AT 1,b;" OPQ": REM esp,(O),(P),(Q)
1060 RETURN
1500 REM rayo
1510 FOR l=1 TO n
1520 INK 0: PLOT 255,170: DRAW -39,-40: DRAW 10,0:
     DRAW -29,-40
1530 POKE 23761,7: RANDOMIZE USR 23760
1540 INK 6: PRINT AT 3,3;"G";AT 4,3 ;"H": REM (G);(H)
1545 PAUSE 5
1550 POKE 23761,1: RANDOMIZE USR 23760
1560 INK 6: PRINT AT 3,3;"G";AT 4,3;"H": REM (G);(H)
1570 INK 0: PLOT 255,170: DRAW -39,-40: DRAW 10,0:
     DRAW -29,-40
1575 PAUSE 1
1580 NEXT l
1590 RETURN
2000 REM fondo
2010 PAPER 4: INK 0: CLS
2020 FOR r=1 TO 100
2030 LET x=INT (RND*255): LET y=INT (RND*80): PLOT x,y
2040 NEXT r: PAPER 5
2050 FOR y=0 TO 12: FOR x=0 TO 31
2060 PRINT AT y,x;" ": REM espacio
2070 NEXT x: NEXT y
2080 INK 0
2090 PLOT 0,72: DRAW 255,0: LET i=4
2100 FOR p=39 TO 0 STEP -2
2110 PLOT 130*SIN (p/30),p: DRAW i,0: LET i=i+2
2120 NEXT p
2130 RETURN
2500 REM castillo
2510 LET x1=10: LET x2=21: LET y1=10: LET y2=16:
     LET lad=400: GO SUB 3000: REM bloque frontal
2520 LET x1=8: LET x2=10: LET y1=6: LET y2=11:
     LET lad=100: GO SUB 3000: REM bloque torre izda.
2530 LET x1=21: LET x2=23: LET y1=6: LET y2=11:
     LET lad=100: GO SUB 3000: REM bloque torre dcha.
```

Continuación Spectrum

```
2540 LET xs=75: LET ys=159:
     GO SUB 3500: REM segmento tejado torre izda.
2550 LET xs=179: LET ys=159:
     GO SUB 3500: REM segmento tejado torre dcha.
2560 FOR r=88 TO 103
2570 PLOT 88,r: DRAW 79,0
2580 NEXT r
2590 FOR b=12 TO 20 STEP 2:
     PRINT AT 11,b;"8": NEXT b: REM (8)
2600 LET x=68: LET y=102: GO SUB 4000
2610 LET x=76: LET y=88: GO SUB 4000
2620 LET x=182: LET y=102: GO SUB 4000
2630 LET x=174: LET y=88: GO SUB 4000
2640 LET x=96: LET y=56: GO SUB 4000
2650 LET x=157: LET y=56: GO SUB 4000
2660 PRINT AT 15,15;"88";AT 16,15;"88":
     REM (8),(8);(8),(8)
2670 RETURN
3000 REM bloque
3010 FOR y=y2 TO y1 STEP -1: FOR x=x1 TO x2
3020 PAPER 2: PRINT AT y,x;" ": REM espacio
3030 NEXT x: NEXT y
3040 LET x1=x1*8: LET x2=x2*8: LET y1=y1*8: LET y2=y2*8
3050 FOR y=y2 TO y1 STEP -2
3060 INK 0: PLOT x1,168-y: DRAW (x2-x1)+7,0
3070 NEXT y
3080 FOR m=1 TO lad
3090 LET a=INT (RND*(x2-x1+7)+x1):
     LET b=168-INT (RND*(y2-y1)+y1): PLOT a,b
3100 NEXT m
3110 RETURN
3500 REM segmento
3510 LET r=39: INK 3
3520 FOR t=160 TO 200
3530 LET a=t*PI/180: LET x=r*SIN (a): LET y=r*COS (a)
3540 PLOT xs,ys: DRAW x,y
3550 NEXT t
3560 RETURN
4000 REM ventana
4010 INK 0: FOR w=y TO y+10: PLOT x,w: DRAW 3,0
4011 NEXT w
4020 RETURN
4500 REM arboles
4510 PAPER 4: INK 0: OVER 1
4520 FOR t=1 TO 30
4530 LET x=INT (RND*29): LET y=13+INT (RND*7)
4540 IF (x>8 AND x<22 AND y<17) THEN  GO TO 4530
4550 PRINT AT y,x;"AB";AT y+1,x;"CD";AT y+2,x;"EF":
     REM (A),(B);(C),(D);(E),(F)
4560 NEXT t
4570 OVER 0: PAUSE 200
4580 RETURN
5000 REM oscuridad
5010 POKE 23761,1: RANDOMIZE USR 23760
5020 PAPER 0: INK 6: PRINT AT 3,3;"G";AT 4,3;"H":
     REM (G);(H)
5030 PAUSE 100
5040 RETURN
5500 REM fuego
5510 DIM f(2,6): INK 6: PAPER 0
5520 FOR g=1 TO 6
5530 READ y,x: LET f(1,g)=y: LET f(2,g)=x
5540 NEXT g
5550 LET n=0
5560 FOR t=1 TO 6
5570 LET n=n+1
5580 FOR d=1 TO 30
5590 FOR j=1 TO n
5600 PRINT AT f(1,j),f(2,j);"T": REM (t)
5610 PRINT AT f(1,j),f(2,j);" ": REM espacio
5620 NEXT j: NEXT d: NEXT t
5630 FOR y=2 TO 11: FOR x=7 TO 23
5640 PRINT AT y,x;" ": REM espacio
5650 NEXT x: NEXT y
5660 RETURN
6000 REM calavera
```

Continuación Spectrum

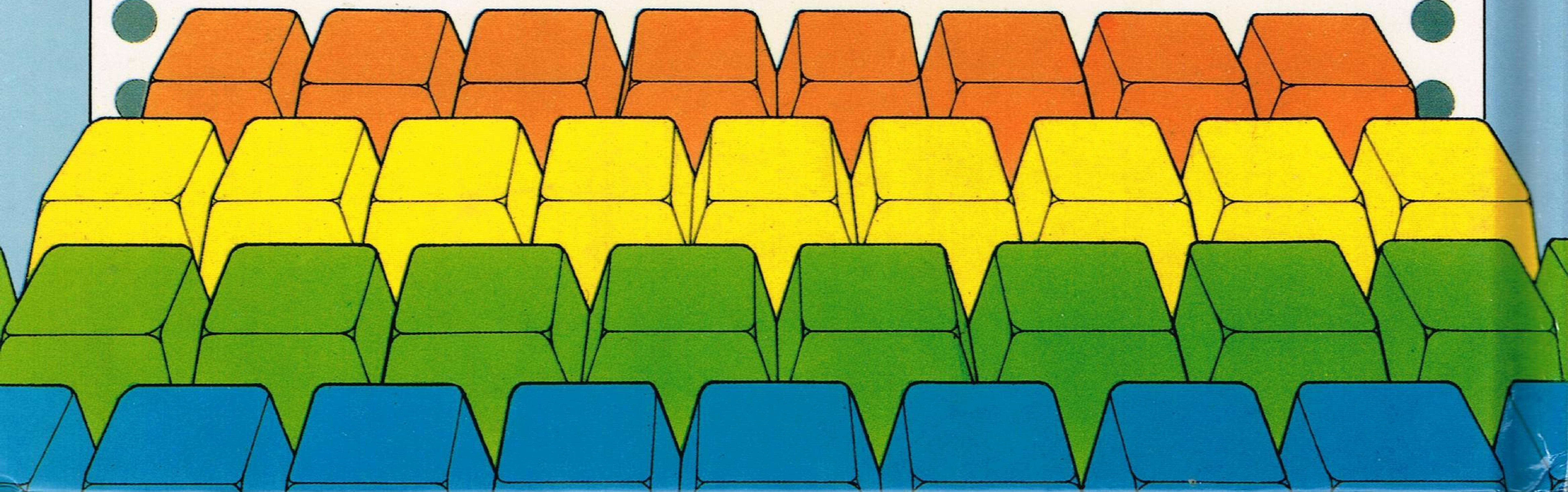
```
6010 INK 0
6020 FOR s=270 TO 450 STEP 2
6030 LET a=s*PI/180: LET x=40*SIN (a)
6040 LET y=40*COS (a): PLOT 127,135: DRAW x,y
6050 NEXT s: LET x1=87: LET x2=80
6060 FOR y=135 TO 87 STEP -1
6070 PLOT x1,y: DRAW x2,0: LET x1=x1+.3: LET x2=x2-.6
6080 NEXT y
6090 PAPER 0: PRINT AT 6,13;"  ";AT 6,17;"  ":
     REM esp,esp;esp,esp
6100 PRINT AT 8,15;"  ": REM espacio,espacio
6110 PRINT AT 10,14;"   ": REM espacio,espacio,espacio
6120 FOR y=81 TO 169 STEP 8: FOR x=87 TO 167 STEP 8
6130 IF POINT (x,y)=1 THEN  PAPER 0:
     INK INT (RND*(7))+1: FLASH 1:
     PRINT AT 22-y/8,x/8-1;"6": FLASH 0: REM (6)
6140 NEXT x: NEXT y
6150 RETURN
6500 REM inicializar
6510 FOR a=65368 TO 65527: READ d: POKE a,d: NEXT a
6520 REM introducir codigo maquina en la sentencia
     REM de la linea 5
6530 FOR i=23760 TO 23777: READ d: POKE i,d: NEXT i
6540 RETURN
7000 DATA 16,129,73,40,16,80,48,9: REM arbol (A)
7010 DATA 128,32,74,134,164,68,204,30: REM arbol (B)
7020 DATA 5,38,178,27,31,15,3,3: REM arbol (C)
7030 DATA 48,96,192,192,194,244,152,128: REM arbol (D)
7040 DATA 1,1,1,3,3,7,31,51: REM arbol (E)
7050 DATA 192,192,192,128,128,128,224,252: REM arbol(F)
7060 DATA 8,16,48,32,96,96,233,224: REM luna (G)
7070 DATA 240,254,127,113,48,54,31,6: REM luna (H)
7080 DATA 1,2,6,230,19,255,17,226: REM bruja (I)
7090 DATA 0,0,0,128,240,128,0,0: REM bruja (J)
7100 DATA 0,0,1,57,4,31,4,56: REM bruja (K)
7110 DATA 64,128,128,160,252,224,64,128: REM bruja (L)
7120 DATA 0,0,0,14,1,7,1,14: REM  bruja (M)
7130 DATA 16,32,96,104,63,248,16,32: REM bruja (N)
7140 DATA 0,0,0,3,0,1,0,3: REM bruja (O)
7150 DATA 4,8,24,154,79,254,68,136: REM bruja (P)
7160 DATA 0,0,0,0,192,0,0,0: REM  bruja (Q)
7170 DATA 0,0,0,66,36,24,24,0: REM murcielago (R)
7180 DATA 0,0,0,0,0,255,24,0: REM murcielago (S)
7190 DATA 8,128,65,2,150,68,40,40: REM llamas (T)
7500 REM datos codigo maquina
7510 DATA 62,0,33,0,88,6,24,197,6,32
7520 DATA 119,35,16,252,193,16,246,201
7530 REM datos posicion ventanas
7540 DATA 10,22,8,23,14,19,14,12,10,9,8,8
```

ESTA NUEVA SERIE CONSTITUYE UNA INTRODUCCIÓN AL ARTE DE PROGRAMAR ORDENADORES. CADA LIBRO ENSEÑA CÓMO ESTRUCTURAR UN PROGRAMA EN RUTINAS, Y AL MISMO TIEMPO EXPLICA Y ANALIZA LO QUE SE LE PIDE AL ORDENADOR QUE HAGA Y POR QUÉ.

¡Y ADEMÁS ES DIVERTIDO! CADA LIBRO CONTIENE COMO MÍNIMO UN JUEGO DE ORDENADOR, APASIONANTE Y ORIGINAL, ESTRUCTURADO DE TAL MODO QUE PUEDE AMPLIARSE O REDUCIRSE, DE ACUERDO CON EL NIVEL O LOS DESEOS DEL USUARIO.

TÍTULOS DE LA SERIE

Gráficos - Iniciación al basic - El banco de datos
Animación - Sonido - Gráficos móviles



APRENDE A PROGRAMAR

ANIMACIÓN

LA CASA ENCANTADA

Marcus Milton

ANAYA